

MOVER 项目文档



小组成员

王静漪 131250116

周睿 131250153

唐珊珊 131250128

王望月 131250173

廖丹琪 131250182

蔡晓莹 131250183

许悠 131250184

王振聪 131250218

2016-3-22

南京大学软件学院

目录

1. 系统主要功能点.....	2
2. P2P 架构设计与评估	2
2.1. ADD 过程	2
2.1.1. 第一次迭代.....	2
2.1.2. 第二次迭代.....	8
2.1.3. 第三次迭代.....	9
2.1.4. 第四次迭代.....	12
2.1.5. 第五次迭代.....	16
2.2. ATAM 评估过程	18
2.2.1. 质量属性效用树.....	18
2.2.2. 架构方法分析.....	19
2.2.3. 敏感点列表.....	23
2.2.4. 权衡点列表.....	23
2.2.5. 有风险决策列表.....	24
2.2.6. 无风险决策列表.....	24
2.3. 架构图.....	24
3. Client/Server 架构设计与评估	26
3.1. ADD 过程	26
3.1.1. 第一次迭代.....	26
3.1.2. 第二次迭代.....	27
3.1.3. 第三次迭代.....	31
3.1.4. 第四次迭代.....	33
3.1.5. 第五次迭代.....	34
3.2. ATAM 评估过程	37
3.2.1. 质量属性效用树.....	37
3.2.2. 架构方法分析.....	38
3.2.3. 敏感点列表.....	43
3.2.4. 权衡点列表.....	44
3.2.5. 有风险决策列表.....	44
3.2.6. 无风险决策列表.....	44
3.3. 架构图.....	45
4. P2P 与 C/S 架构对比.....	47
4.1. 性能对比.....	47
4.2. 安全性对比.....	47
4.3. 资源利用对比.....	47
5. 系统类图设计.....	47
6. 挑战和经验.....	49

7. 组员和分工.....	49
---------------	----

1. 系统主要功能点

- 1) 健身教学：用户通过在线直播观看教练的课程，达到远程随时随地在健身房的效果。
- 2) 健身视频：用户通过互联网观看过往健身课程视频。
- 3) 健康管理：用户管理个人的每日运动、身体、睡眠状况，可以进行上传数据、设定运动目标、设置身体数据、查看每日健康情况的操作。可以实现可穿戴设备的数据采集。
- 4) 用户管理：管理员可以对账户进行增删改查，普通用户可以注册，修改个人账户信息。
- 5) 权限管理：基于角色的权限管理，个人用户、教练、医生、系统管理员。
- 6) 建议管理：教练、医生针对用户建议。
- 7) 统计分析：用户或者其对应的教练和医生，可以通过查看对历史数据统计分析的展示结果来回顾个人健康历程，从而进一步帮助后期计划。
- 8) 建议导入：支持 xml、excel 文件格式的信息导入，支持批量导入。

2. P2P 架构设计与评估

2.1. ADD 过程

2.1.1. 第一次迭代

● 步骤一

获得样本输入，筛选掉不必要的需求，列出 ASR：

可维护性：场景 1:手环升级或需要连接新设备

场景组成部分	可能的值
源	开发人员
刺激	手环升级或添加新的连接设备
制品	硬件连接模块，文件传输模块
环境	系统在构建中或系统已上线运行
响应	修改硬件连接模块和文件传输模块相应代码 测试硬件连接模块和文件传输模块 发布维护变更
响应度量	维护时间不应超过 3 小时 维护所影响到的代码量不能超过 2%

可维护性：场景 2：操作系统升级

场景组成部分	可能的值
源	操作系统
刺激	操作系统升级，有新特性
制品	系统
环境	系统开发时，运行时，维护时
响应	修改与操作系统接口直接相关的代码 测试系统功能 系统能在新的操作系统上运行并保持原有功能实现
响应度量	系统的维护修改能够在 0.5 个人月内完成 维护所影响到的代码量不能超过 5% 系统完成维护修改所需成本不超过系统开发总成本的 5%

可维护性：场景 3：建议文件格式增加

场景组成部分	可能的值
源	开发人员
刺激	要支持新的建议文件格式
制品	文件传输模块
环境	系统在构建中或系统已上线运行
响应	修改文件传输模块相应代码 测试文件传输模块 发布维护变更
响应度量	维护时间不应超过 2 小时 维护所影响到的代码量不能超过 1%

易用性：场景 4：熟练度培训

场景组成部分	可能的值
源	终端用户
刺激	使用该系统的直播拍摄相关工作人员，需要通过培训熟练掌握直播功能的使用
制品	系统
环境	运行时或配置时
响应	帮助系统与拍摄环境联系紧密 界面为用户所熟悉，支持界面的有效导航 提供使用手册 支持用户提出使用中遇到的问题，记录并存储其内容
响应度量	对系统满意的用户数达到总人数的 70%

	用户熟练掌握该功能所需时间在一周以内 在培训期间解决的问题数量达到请求总数的 85%
--	---

易用性：场景 5：正常操作

场景组成部分	可能的值
源	终端用户
刺激	用户执行一项操作，如上传可穿戴设备数据、查看历史统计数据、观看健身课程直播等操作
制品	系统
环境	运行时
响应	正确引导用户操作步骤 高效且成功地完成该项操作 从错误中及时恢复，防止数据丢失
响应度量	完成单项细节操作的时间不多于 2 分钟 操作成功率高于 99% 错误发生后恢复到重新正常使用所需时间少于 10 分钟 错误发生后数据丢失率低于 1%

可伸缩性：场景 6：用户数量增加

场景组成部分	可能的值
源	终端用户
刺激	同时在线观看视频直播的用户数量增加
制品	系统
环境	系统已上线运行
响应	正常进行视频直播
响应度量	可支持的并发用户数量不低于 10000 请求观看直播成功的用户数量不低于 95% 用户请求直播连接的平均请求时间不超过 5 秒

安全性 场景 7：未经授权的访问

场景组成部分	可能的值
源	未经授权的个人或其它系统
刺激	试图显示数据或改变/删除数据，试图改变系统服务
制品	系统服务，系统中的数据
环境	系统联网状态下运行
响应	识别用户是否已经授权 阻止对服务或数据的访问 以一种不可读的方式存储数据

	数据或服务遭到破坏后可恢复
响应度量	成功识别攻击的概率不低于 99% 在拒绝服务的情况下，攻击者仍能获得服务的概率低于 1% 恢复数据和服务可在 3 人日内完成

安全性 场景 8：经过授权的访问

场景组成部分	可能的值
源	经过授权的个人或系统
刺激	试图访问不在其权限范围内的数据或服务
制品	系统服务，系统中的数据
环境	系统联网状态下运行
响应	对用户的身份进行验证 允许访问用户权限范围内的服务和数据 拒绝访问不在用户权限范围内的服务和数据，并通知用户
响应度量	成功识别用户身份的概率不低于 99% 拒绝不合法访问的概率高于 99.9% 在拒绝服务的情况下，用户仍能获得服务的概率低于 0.1%

性能 场景 9：客户端进行网络访问，如提出问题、观看直播和视频、数据同步等

场景组成部分	可能的值
源	终端用户
刺激	用户使用客户端进行网络访问
制品	服务器、客户端中涉及到网络访问的模块
环境	系统运行时
响应	良好的网络访问
响应度量	服务器带宽至少为 1Gbps 系统可以保证同时响应 10000 名用户的操作 客户端向服务器/客户端发出一次请求后的响应时间应少于 2s 客户端请求直播和视频数据的缓冲时间应不大于 10s

性能 场景 10：正常使用客户端

场景组成部分	可能的值
源	终端用户
刺激	用户正常使用客户端
制品	客户端
环境	系统运行时
响应	良好的用户操作

响应度量	客户端中用户操作的响应时间应小于 1s
------	---------------------

可用性 场景 11：服务失效需要修复

场景组成部分	可能的值
源	维护人员
刺激	服务失效需要修复
制品	重启的可用的服务
环境	系统正在运行
响应	查明服务失效原因 修复失效部分 重启服务
响应度量	服务 24 小时可用 服务失效修复时间<5 分钟

可用性 场景 12：直播数据流中断

场景组成部分	可能的值
源	维护人员
刺激	直播数据流中断
制品	可用的直播
环境	系统正在运行
响应	查明数据流中断原因（是推流还是拉流的问题） 更换数据流源或重载数据流
响应度量	数据流源更换时间<10 秒 数据流源修复时间<10 分钟

可扩展性 场景 13：需要增加一个新的功能

场景组成部分	可能的值
源	开发人员
刺激	需要增加一个新的功能
制品	新的功能模块
环境	系统在构建中或系统已上线运行
响应	编写新功能代码，修改原有代码 测试代码 发布维护变更
响应度量	维护时间不应超过 3 小时 维护所影响到的原代码量不能超过 2%

互操作性 场景 14：需要与可穿戴设备进行交互

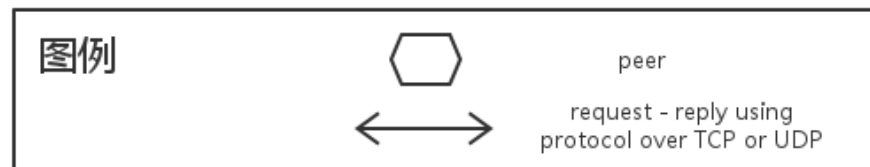
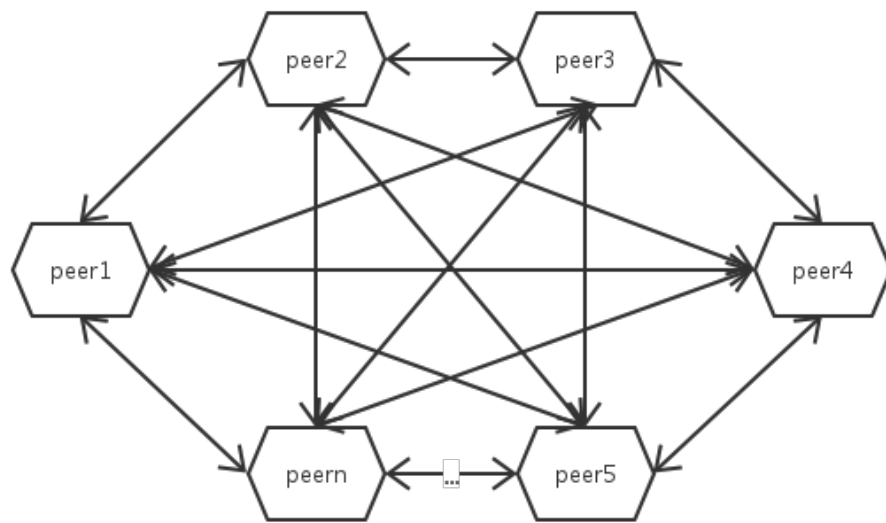
场景组成部分	可能的值
源	开发人员
刺激	希望现有系统能与一个特定可穿戴设备进行交互
制品	系统
环境	系统运行时或构建时
响应	与指定可穿戴设备进行数据传输
响应度量	交互成功率大于 95%

可移植性 场景 15：服务器或客户端移植到其他环境

场景组成部分	可能的值
源	开发人员、维护人员
刺激	服务器或客户端移植到其他运行环境
制品	服务器、客户端
环境	开发时、运行时或维护时
响应	完成移植所需的修改 测试修改部分 部署系统到新的环境中
响应度量	完成移植所花费的代价不得高于 4 人/月 度量 $DP > 0.7$ ($DP = 1 - \text{软件移植成本估计} / \text{软件重新开发成本估计}$)

● 步骤二

对整个系统进行分解，分为对等的各个节点。



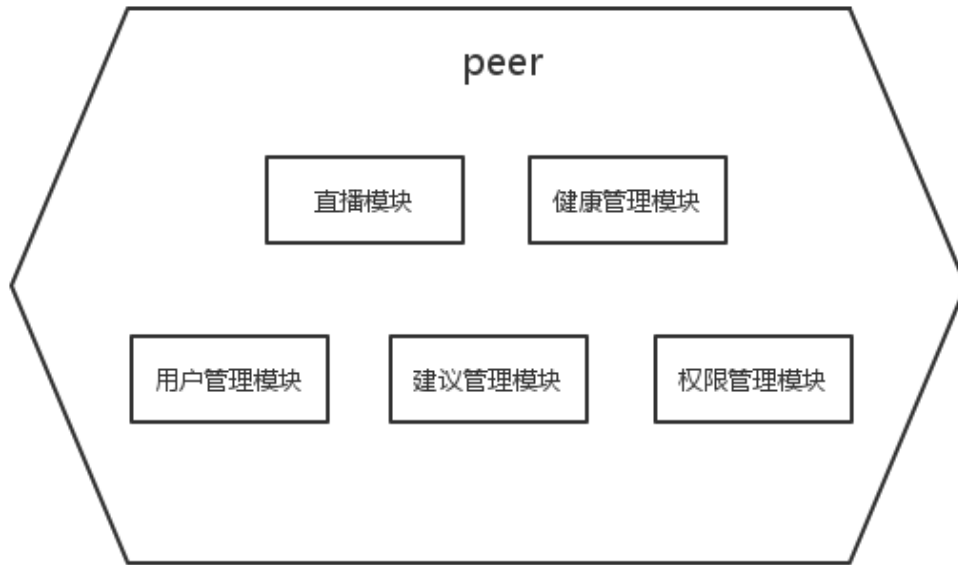
2.1.2. 第二次迭代

- 步骤一

第二次迭代可以不用此步骤。

- 步骤二

对单个节点进行分解 ,并分为直播模块、健康管理模块、用户管理模块、建议管理模块、权限管理模块。



2.1.3. 第三次迭代

- 步骤一

第三次迭代可以不用此步骤。

- 步骤二

在此次迭代中针对视频播放模块进行分解。

- 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 6：用户数量增加	中	高
2	场景 9：客户端进行网络访问，如提出问题、观看直播和视频、数据同步等	高	高
3	场景 11：服务失效需要修复	高	中
4	场景 12：直播数据流中断	中	中
5	场景 15：服务器或客户端移植到其他环境	中	中

- 步骤四

选择满足架构驱动因素的架构模式。

- 1) 针对性能相关的架构驱动因素

- i. 设计关注点

设计关注点	子关注点
直播观看体验	直播清晰流畅延迟短
	可支持较多用户同时观看直播
视频播放体验	播放清晰流畅延迟短

ii. 各子关注点的候选架构模式

a) 直播清晰流畅不卡顿

#	模式名称	开发时间开销	成本开销	带宽要求
1	提高 CPU 利用率	高	低	不要求
2	小包传输	中	中	低
3	动态路由优化	中	高	中

✧ 选择理由：提高 CPU 利用率实现难度较大，不列入选择。小包传输和动态路由优化两者在开发时间与成本开销上各有优劣，且没有显著缺点，因此都作为备选方案。

b) 可支持较多用户同时观看直播

#	模式名称	开发时间开销	成本开销	系统资源开销
1	CDN (内容分发网络)	中	中	中
2	使用第三方软件管理负载均衡	中	中	高
3	使用硬件管理负载均衡	低	高	低

✧ 选择理由：硬件管理负载均衡成本过高，不列入选择。CDN 与软件管理负载均衡在开发时间、成本与资源方面开销差距不大，因此都作为备选方案。

c) 播放清晰流畅不卡顿

#	模式名称	开发时间开销	成本开销	数据丢失风险
1	缓存	中	中	无
2	数据库读写分离+分区	高	中	有

✧ 选择理由：采用缓存与数据库读写分离+分区都是技术较为成熟的常用方案，因此都作为备选。

iii. 候选架构模式综合评估

	直播清晰流畅不卡顿		可支持较多用户同时观看直播		播放清晰流畅不卡顿	
	优点	缺点	优点	缺点	优点	缺点
小包传输	带宽要求低	网络利用率低，有额外资源开销	无显著影响		带宽要求低	网络利用率低，有额外资源开销
动态路由优化	效率高	消耗额外	无显著影	容易造成	效率高	消耗额外

		计算资源	响	拥堵		计算资源
CDN	效率高	无显著缺点	解决 ISP 互通问题和网络链路问题 ;减轻了源服务器的压力 ;有效抗 DDOS 攻击		效率高	容量小
第三方软件管理负载均衡	使用灵活且可以满足普通需求	软件运行消耗资源	成本低廉 , 配置简单	当连接数量大时 , 性能受到软件本身制约	成本低廉又可以满足普通需求	软件运行消耗资源
缓存	无显著影响		无显著影响		带宽要求低	无显著缺点
数据库读写分离+分区	提高读的效率	可能会造成直播故障	读资源丰富	修复部分故障时间长 , 影响大	读数据效率高	有数据丢失风险

✧ 评估结果 : 动态路由优化在性能提升方面比小包传输更有优势 ;CDN 是更完整的解决方案 , 包含一部分负载均衡作用 ;读写分离 : 在主从数据库复制数据时可能造成数据丢失 , 存在风险。

2) 针对可用性相关的架构驱动因素

i. 设计关注点

设计关注点	子关注点
直播观看体验	直播稳定不中断
视频播放体验	视频播放稳定不中断

ii. 各子关注点的候选架构模式

a) 直播稳定不中断

#	模式名称	配置难度	稳定性	网络依赖
1	采用 Nginx	配置简单	较低	低
2	采用 LVS 集群	配置较难	高	高

✧ 选择理由 : Nginx 与 LVS 都是软件实现负载均衡的常用方案 , 因此在配置性和稳定性上各有优劣 , 因此都作为备选方案。

b) 视频播放稳定不中断

#	模式名称	系统资源开销	失效恢复时间
1	路由按需恢复	低	较长
2	路由稳定化恢复	高	短

- ✧ 选择理由：路由按需恢复开销小，能适应普通的扰动，但难以应对有大量的结点频繁并且并发的加入、离开或者失效的高扰动环境；稳定化恢复相对开销大，但对于高扰环境适应性好，因此作为备选方案。

iii. 候选架构模式综合评估

	直播稳定不中断		视频播放稳定不中断	
	优点	缺点	优点	缺点
采用 Nginx	性能好；功能多，除了负载均衡，还能作 Web 服务器；社区活跃，第三方补丁和模块很多	不支持 session 保持；对后端 real server 的健康检查功能效果不好	性能好；功能多	负载度和稳定度相对较低；无现成的双机热备方案
采用 LVS 集群	抗负载能力强；工作稳定，自身有完整的双机热备方案；应用范围较广	配置困难；操作人要求高；健康检测需要另外配置 Ldirector	抗负载能力强；工作稳定	配置困难
路由稳定化恢复	无显著影响		对于高扰动环境适应性好	需要一定资源开销

- ✧ 评估结果：路由稳定化恢复是视频播放稳定不中断这一子关注点的备选架构模式，且对直播稳定也无显著影响，因此作为最终决策。与 Nginx 相比，LVS 的负载度和稳定度更高，对于直播和视频的稳定播放来说更为可靠，同时也有利于应对高并发场景，因此采用 LVS 为最终决策。

2.1.4. 第四次迭代

● 步骤一

第四次迭代可以不用此步骤。

● 步骤二

在此次迭代中针对健康管理播放模块进行分解。

● 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 1：手环升级或需要连接新设备	中	中
2	场景 2：操作系统升级	高	中

3	场景 3：建议文件格式增加	低	低
4	场景 14：需要与可穿戴设备进行交互	高	中
5	场景 15：服务器或客户端移植到其他环境	中	中

● 步骤四

选择满足架构驱动因素的架构模式。

1) 针对可维护性相关的架构驱动因素

i. 设计关注点

设计关注点	子关注点
环境升级	可穿戴设备升级
	操作系统升级
需求改变	文件格式改变

ii. 各子关注点的候选架构模式

a) 可穿戴设备升级

#	模式名称	开发时间开销	成本开销	对其他模块影响
1	封装可穿戴设备相关代码模块 + 添加接口	中	中	低
2	封装可穿戴设备相关代码模块 + 适配器模式	中	中	无影响

选择理由：模式 2 相比模式 1，可以使用同一个接口完成不同类型设备的升级，更好的隐藏了与可穿戴设备交互的信息，所以选择模式 2。

b) 操作系统升级

#	模式名称	开发时间开销	成本开销	对所有模块影响程度
1	封装操作系统变更更容易影响的相关代码	中	中	低
2	使用与平台无关的开发语言	中	高	无

◇ 选择理由：模式 2 是一种很好的想法，但是针对本次项目，iOS 开发语言的限制导致其更新换代势必会影响到 API，所以使用模式 1。

c) 文件格式改变

#	模式名称	时间开销	成本开销	对其他模块的影响程度
1	预内置支持多种主流格式	不稳定	不稳定	高
2	良好封装文件格式处理模	稳定中等	稳定中等	低

	块，保证接口稳定			
3	数据标准化，选择标准的数据交换方式作为中介	稳定较低	稳定中等	无影响

◇ 选择理由：文件格式发展难以很好的预测，所以模式 1 不是非常有效。模式 2 中的文件格式接口承载了过大的修改压力，而模式 3 相比模式 2 可以更加灵活的应对文件格式改变，只要开发新文件格式与中介格式转化的代码即可，不论是输入还是输出的文件格式变化，都不会影响到其他代码，易于管理。所以选择模式 3。

iii. 候选架构模式综合评估

	可穿戴设备升级		操作系统升级		文件格式变化	
	优点	缺点	优点	缺点	优点	缺点
封装可穿戴设备相关代码模块 + 适配器模式	隐藏设备信息,对其他模块影响小	无显著缺点	无显著影响	无显著影响	文件格式变化不会影响到其他模块	无显著缺点
封装易受操作系统影响代码	增强了可扩展性	无显著缺点	降低了操作系统影响范围	难以界定受影响的代码范围	无显著影响	无显著影响
数据标准化	保证了数据传递稳定	设备更新换代需要开发相应转换代码	无显著影响	无显著影响	隐藏了文件格式转换信息,灵活	无显著缺点

◇ 评估结果：被选择的三个模式互相之间没有显著矛盾，可以作为最终方案。

2) 针对互操作性相关的架构驱动因素

i. 设计关注点

设计关注点	子关注点
可穿戴设备的数据采集	及时采集数据
	保证数据安全完整

ii. 各子关注点的候选架构模式

a) 及时采集数据

#	模式名称	开发时间开销	可处理的业务逻辑	及时性
1	使用数据库触发器及时处理数据变动	低	简单	高
2	交换代理程序监控数据变动	高	复杂	低

- ◇ 选择理由：这两种模式都是技术上较为成熟的常用解决方案，且各有优劣，都作为备选方案。

b) 保证数据安全完整

#	模式名称	数据安全	采集数据的速度	对系统的要求
1	系统接口对接	高	高	适用于新系统的开发
2	数据库中间表	中	中	适用于不能修改程序代码的老系统
3	交换文件	中	低	适用于无法提供中间表的情况

- ◇ 选择理由：由于我们是开发一个新系统而不是在老系统的基础上修改，所以选择在安全性和处理速度方面都具有优势的系统接口对接模式。

iii. 候选架构模式综合评估

	及时采集数据		保证数据安全完整	
	优点	缺点	优点	缺点
系统接口对接	针对接口开发，可以及时获取新的数据	无显著缺点	性能好，数据传输稳定。减少了读写中间表或中间文件的次数，数据不易出错或丢失	需要外部系统有较为稳定的开发接口
使用数据库触发器	使用数据库触发器机制可及时，快速应对数据变动 开发时间短，成本低	无法处理过于复杂的逻辑	数据库操作可以使用事务来保证数据的完整性	无显著缺点
交换代理程序	可以较为及时的处理数据变动，交换代理程序可以处理复杂业务逻辑	依靠代理程序监控文件，及时性不足 开发时间长，成本高	无显著影响	开发时间长，成本高

- ◇ 评估结果：系统接口对接是保证数据安全完整这一子关注点的备选架构模式，且对及时采集数据也十分有利，因此作为最终决策。我们的系统中对可穿戴设备的数据主要是采集和简单统计，不需要过于复杂的业务逻辑，使用数据库触发器处理数据更及时，成本较低，同时对数据完整性有较为有利，因此采用数据库触发器为最终决策。

2.1.5. 第五次迭代

- 步骤一

第五次迭代可以不用此步骤。

- 步骤二

在此次迭代中针对视频播放模块进行分解。

- 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 5：正常操作	高	中
2	场景 7：未经授权的访问	高	高
3	场景 8：经过授权的访问	高	中
4	场景 11：服务失效需要修复	高	中

- 步骤四

选择满足架构驱动因素的架构模式。

1) 针对安全性相关的架构驱动因素

i. 设计关注点

设计关注点	子关注点
机密性	阻止未经授权的访问
	视频资源传输安全
完整性	拒绝用户访问权限以外数据

ii. 各子关注点的候选架构模式

a) 阻止未经授权的访问

#	模式名称	开发时间开销	成本开销	局限性	额外资源开销
1	基于信任和名誉的双层认证机制	高	中	无	中
2	第三方信任认证机制	中	中	第三方易被攻击	低
3	防火墙	低	低	内外网连接不安全	低

✧ 选择理由：由于 P2P 架构的特殊性，我们对于安全性要求比较高，所以即使会产生一部分额外的计算、通信资源开销，我们选择基于信任和名誉的双层认证机制作为备选方案。

b) 视频资源传输安全

#	模式名称	可实施性	可管理性	灵活性
1	网络层安全隧道	高	中	中
2	传输层安全隧道 (SSL)	中	高	高
3	应用层安全隧道	中	高	高

✧ 选择理由：在视频资源传输中，对于其版权保护要求高，而这三种方案从三个层次进行不同实现，因此暂时都保留作为备选方案。

c) 拒绝用户访问权限以外数据

#	模式名称	开发时间开销	成本开销	技术难度
1	使用 JXTA 开发平台	中	高	低
2	基于角色管理的认证机制	中	低	中

✧ 选择理由：由于 JXTA 开发平台作为 P2P 架构来说已经比较稳定，技术难度低，因此两个方案都作为备选方案。

iii. 候选架构模式综合评估

	阻止未经授权的访问		视频资源传输安全		拒绝用户访问权限以外数据	
	优点	缺点	优点	缺点	优点	缺点
基于信任和名誉的双层认证机制	安全性高	有额外的计算、存储资源开销	对安全性有帮助	无显著缺点	对安全性有帮助	无显著缺点
网络层安全隧道	防止验证信息被篡改	无显著缺点	可实施性高	要求节点都支持 IPSec，不能穿越 NAT	无显著影响	
传输层安全隧道 (SSL)	防止验证信息被篡改	无显著缺点	可管理性强，灵活性高	无显著缺点	无显著影响	
应用层安全隧道	防止验证信息被篡改	无显著缺点	灵活性高	实现复杂度高	无显著影响	
使用 JXTA 开发平台	提供安全协议	无显著缺点	提供安全协议	无显著缺点	技术成熟，实现复杂度低	架构单一
基于角色管理	无显著影响		提高安全	无显著缺点	灵活性高	比较基

的认证机制		性	点		本，没有特殊优势
-------	--	---	---	--	----------

- ◇ 评估结果 :我们选择基于信任和名誉的双层认证机制来作为阻止未经授权的访问这一关注点的最终决策，是考虑到这种机制最为安全同时资源开销也在可承受范围内。由于安全隧道技术，实现层次越高，则对网络设施的依赖性越小，灵活性更好，但实现复杂度更高。而我们的系统是初次开发，没有修改成本，不需要特别高的灵活度，所以我们折衷选择传输层安全隧道技术。JXTA 开发平台是 P2P 技术成熟的标准组件平台，开源代码成本低，因此作为拒绝用户访问权限外数据这一子关注点的最终选择。

2.2. ATAM 评估过程

2.2.1. 质量属性效用树

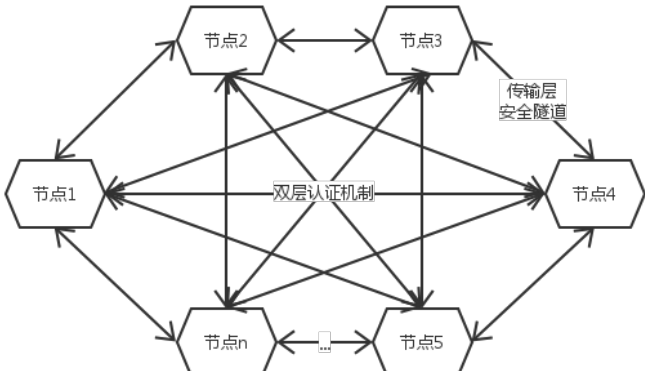
质量属性	属性求精	场景
性能	直播响应时间	客户端向服务器/客户端发出一次请求后的响应时间应少于 2s (M , M)
		客户端请求直播和视频数据的缓冲时间应不大于 10s (M , M)
	吞吐量	在系统正常运行的状态下，系统可以保证同时响应 10000 名用户的操作 (H , H)
可用性		系统服务 24 小时可用 (H , M)
		直播数据流中断时，数据流源修复时间 < 10 分钟 (M, M)
可伸缩性		在系统正常运行状态下，可支持的并发用户数量不低于 10000 (H , M)
		用户请求直播连接的平均请求时间不超过 5 秒 (H , M)
安全性	机密性	系统验证用户身份，拒绝不在用户权限范围内的服务和数据访问，并通知用户 (H , H)
	完整性	系统拒绝未经授权的访问 (H , H)
可维护性		当可穿戴设备升级时，维护相应代码的时间应不超过 3 小时，并且受影响的代码量小于 2% (H , M)
		当操作系统的新特性影响到系统原有功能的实现时，系统应该能在 0.5 个人月的成本下完成更新，并且成本不应超过总开发成本的 5% (H , M)
		当增加新的建议文件格式时，维护相应代码的时间应不超过 2 小时，并且受影响的代码量小于 1%

		(H , M)
可移植性		服务器或客户端移植到其他环境，完成移植所花费的代价不得高于 4 人/月 (H , H)
互操作性	数据接口	本系统可与指定可穿戴设备进行数据传输，交互成功率大于 95% (H , H)
易用性	正常操作	用户成功执行一项细节操作，如上传可穿戴设备数据、查看历史统计数据、观看健身课程直播等操作，完成单项操作的时间不超过 2 分钟 (H , H)
	熟练度培训	使用该系统的直播拍摄相关工作人员，掌握直播功能操作所需时间在一周以内 (H , M)

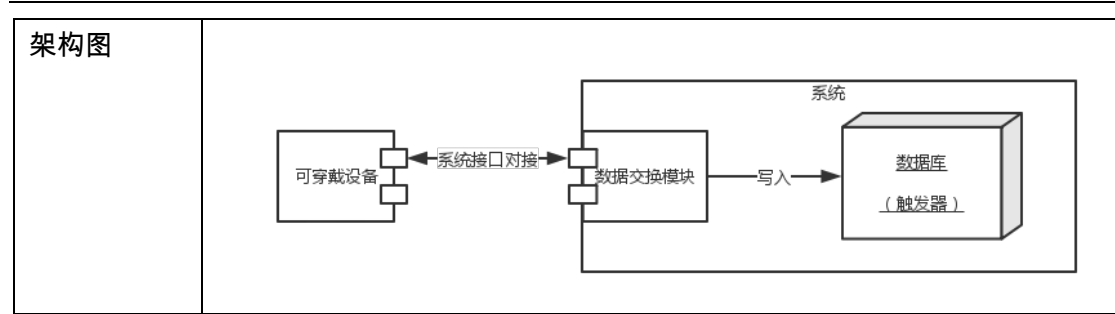
2.2.2. 架构方法分析

场景号：A1	场景：用户请求直播后获得响应			
属性	性能			
环境	系统正常运行			
刺激	用户请求直播			
响应	获得响应的平均等待时间不大于 2 秒			
架构决策	敏感点	权衡点	有风险决策	无风险决策
小包传输	S1		R1	
动态路由优化	S2			N1
CDN	S3			N2
推理	<p>小包传输对用户带宽要求低，但网络利用率低，有额外资源开销</p> <p>动态路由优化的传输效率高</p> <p>CDN 传输效率高，可解决 ISP 互通问题和网络链路问题，有效抗 DDOS 攻击</p>			
架构图				

场景号：A2	场景：阻止未经授权的访问			
属性	安全性			
环境	系统正常运行			
刺激	试图显示数据或改变/删除数据，试图改变系统服务			
响应	验证用户是否已经授权，阻止对服务或数据的访问			

架构决策	敏感点	权衡点	有风险决策	无风险决策
基于信任和名誉的双层认证机制	S4		R2	
传输层安全隧道 (SSL)	S5			N3
推理	使用基于信任和名誉的双层认证机制安全性高,但有额外的计算、存储资源开销 传输层安全隧道(SSL)可防止验证信息被篡改,且可管理性强,灵活度高			
架构图				

场景号：A3	场景：接收手环传输的数据			
属性	互操作性			
环境	系统运行时或构建时			
刺激	希望现有系统能与一个特定可穿戴设备进行交互			
响应	与指定可穿戴设备进行数据传输，交互成功率大于 95%			
架构决策	敏感点	权衡点	有风险决策	无风险决策
系统接口对接	S6			N4
使用数据库触发器	S7			N5
推理	系统接口对接针对接口开发，性能好，数据传输稳定。减少了读写中间表或中间文件的次数，数据不易出错或丢失 使用数据库触发器机制可及时，快速应对数据变动，开发时间短，成本低			

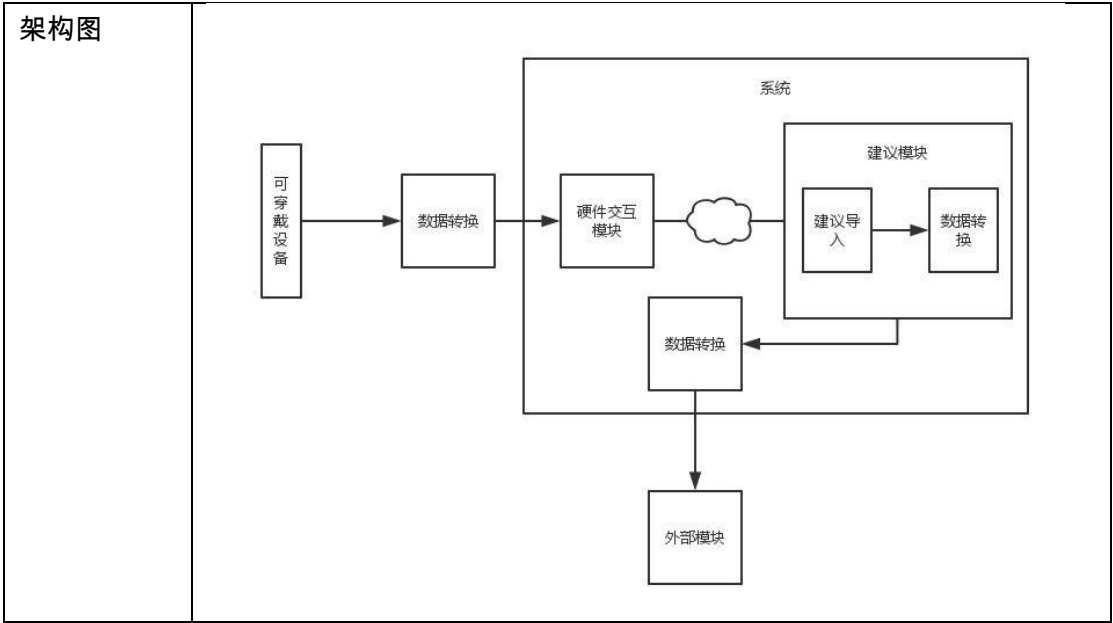


场景号：A4	场景：直播数据流中断后及时修复			
属性	可用性			
环境	系统运行时			
刺激	直播数据流中断			
响应	直播数据流源修复时间<10 分钟			
架构决策	敏感点	权衡点	有风险决策	无风险决策
采用 LVS 集群	S8	T1 (可配置性)	R3	
路由稳定化恢复	S9	T2 (性能)	R4	
推理	<p>LVS 集群在负载均衡软件中抗负载能力最强，且工作稳定，应用范围较广，但配置较困难，对于操作人要求高</p> <p>路由稳定化恢复策略对路由表进行周期性检测并加以修复，适应于 P2P 架构下节点变化大的高扰动环境，但资源开销相对较大，一定程度上影响性能</p>			
架构图				

场景号：A5	场景：可穿戴设备升级			
属性	可维护性			
环境	系统运行时或构建时			
刺激	手环升级或添加新的连接设备			
响应	维护并发布的时间小于 3 小时且受影响代码小于 2%			
架构决策	敏感点	权衡点	有风险决策	无风险决策

适配器模式	S10			N6
封装可穿戴设备相关代码模块	S11			N7
推理	适配器模式隐藏了硬件接口,其变更不会影响外部模块,提升了可维护性。封装可穿戴设备相关代码将变更控制在一定范围内,提升了可维护性。			
架构图	<pre> graph LR subgraph WDM [可穿戴设备模块] subgraph HIM [硬件交互模块] direction TB II[交互实现] AD[适配器] II --> AD end AD --> I[<<interface>> getStat] end WE[可穿戴设备] --> WDM OM[其他模块] --> I </pre>			

场景号：A6	场景：增加文件格式			
属性	可维护性			
环境	系统运行时或构建时			
刺激	文件格式更改或要添加新的文件格式			
响应	维护并发布的时间小于 3 小时且受影响代码小于 2%			
架构决策	敏感点	权衡点	有风险决策	无风险决策
数据标准化	S12	T3 (性能)		N8
封装文件处理模块	S13			N9
推理	数据标准化使用中介格式,灵活应对文件格式变化,只要开发新文件格式与中介格式转化的代码即可,不论是输入还是输出的文件格式变化,都不会影响到其他代码,易于管理。多次数据格式处理降低了性能。封装文件处理模块使得变更控制在了一定范围内,提升了可维护性。			



2.2.3. 敏感点列表

编号	架构决策	相关质量属性	场景编号
S1	小包传输	性能	A1
S2	动态路由优化		
S3	CDN		
S4	基于信任和名誉的双层认证机制	安全性	A2
S5	传输层安全隧道 (SSL)		
S6	系统接口对接	互操作性	A3
S7	使用数据库触发器		
S8	采用 LVS 集群	可用性	A4
S9	路由稳定化恢复		
S10	适配器模式	可维护性	A5
S11	封装可穿戴设备相关代码模块		
S12	数据标准化		A6
S13	封装文件处理模块		

2.2.4. 权衡点列表

编号	架构决策	相关质量属性	场景编号
T1	采用 LVS 集群	可用性，可配置性	A4
T2	路由稳定化恢复	可用性，性能	A4
T3	数据标准化	可维护性，性能	A6

2.2.5. 有风险决策列表

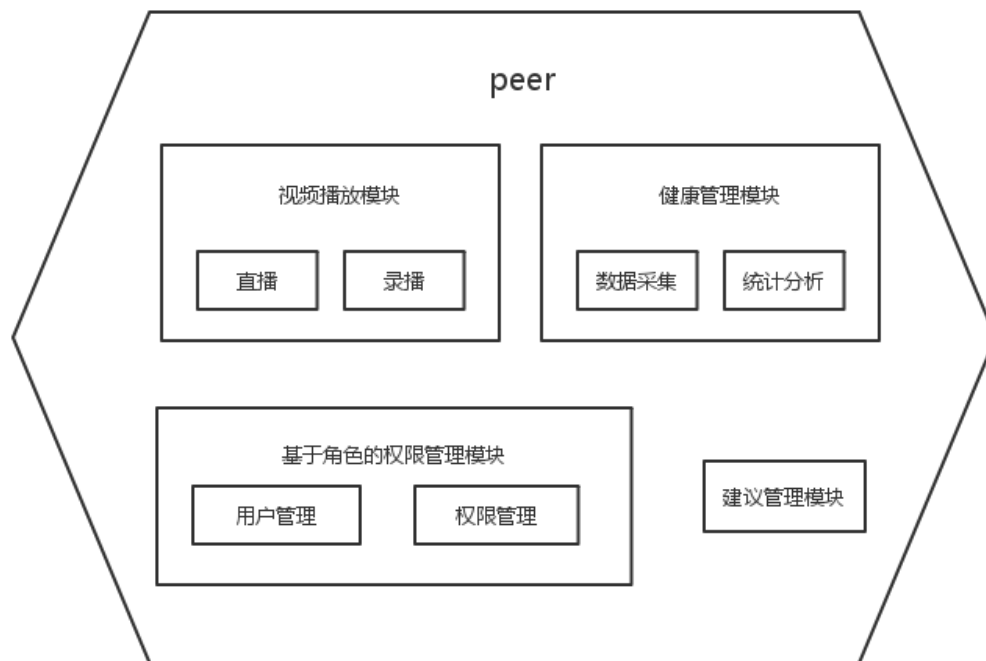
编号	架构决策	风险	场景编号
R1	小包传输	可能影响其他功能的响应时间	A1
R2	基于信任和名誉的双层认证机制	有额外的计算、存储资源开销	A2
R3	采用 LVS 集群	配置较困难，对于操作人要求高	A4
R4	路由稳定化恢复	资源开销相对较大，一定程度上影响性能	A4

2.2.6. 无风险决策列表

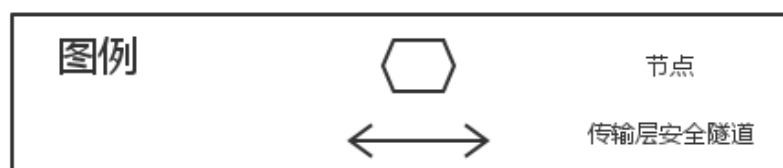
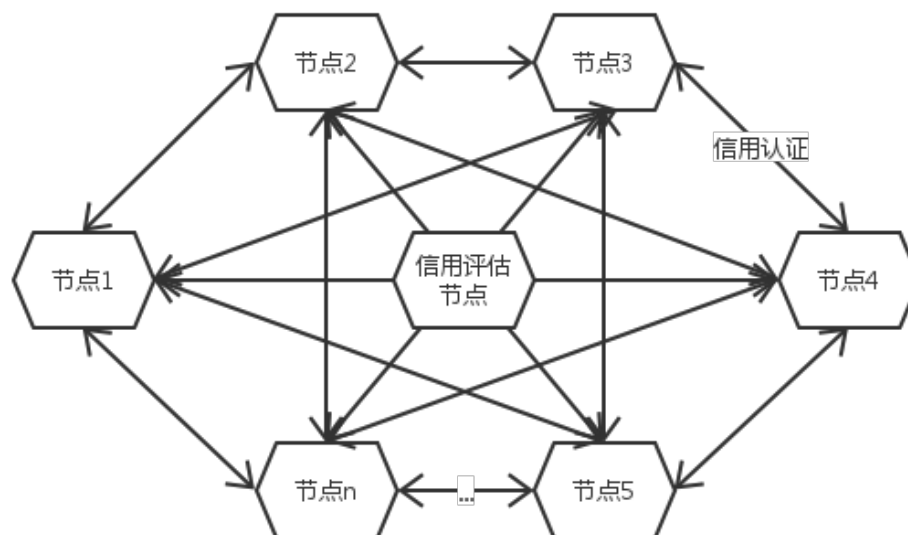
- N1: 动态路由优化。
- N2: 采用 CDN
- N3: 传输层安全隧道 (SSL)。
- N4: 系统接口对接
- N5: 使用数据库触发器。
- N6: 适配器模式。
- N7: 封装与操作系统相关的代码。
- N8: 数据标准化。
- N9: 封装文件处理模块。

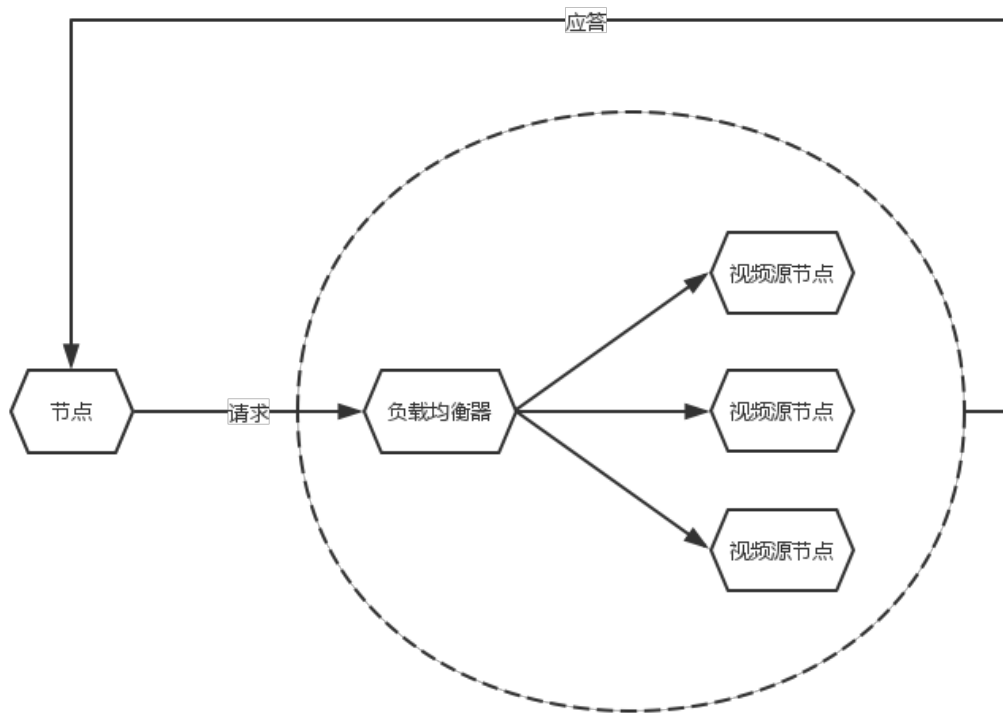
2.3. 架构图

1) 模块视图



2) C&C 视图





3. Client/Server 架构设计与评估

3.1. ADD 过程

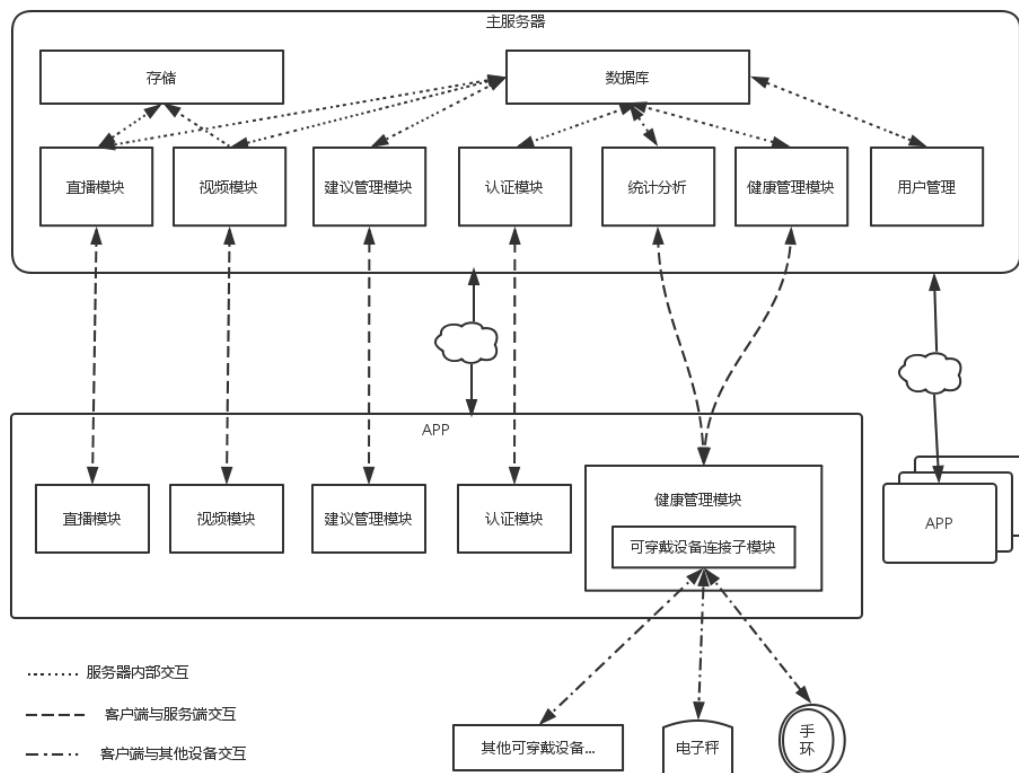
3.1.1. 第一次迭代

- 步骤一

获得样本输入，筛选掉不必要的需求，列出 ASR。此处的 ASR 与 P2P 架构设计 ADD 过程中所列相同。

- 步骤二

对整个系统进行分解，得到系统组成元素视图如下：



3.1.2. 第二次迭代

● 步骤一

第三次迭代可以不用此步骤。

● 步骤二

我们选择系统 (服务器和客户端) 的直播模块进行分解。直播模块是系统的核心功能模块之一，它可以分为直播源采集 (客户端)、直播数据传输 (客户端和服务端之间)、直播数据处理 (服务器) 和直播观看 (客户端) 四个分解功能点。对直播模块进行分解，对于进一步设计该系统的体系结构有很大的帮助。

● 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 2：操作系统升级	高	中
2	场景 4：熟练度培训	高	低
3	场景 5：正常操作	高	中
4	场景 6：用户数量增加	中	高
5	场景 9：客户端进行网络访问，如提出问题、观看	高	高

	直播和视频、数据同步等		
6	场景 11：服务失效需要修复	高	中
7	场景 12：直播数据流中断	中	中
8	场景 13：需要增加一个新的功能	高	中
9	场景 15：服务器或客户端移植到其他环境	中	中

● 步骤四

选择满足架构驱动因素的架构模式。

i. 设计关注点

设计关注点	子关注点
直播源采集	直播源采集方式
	直播源压缩编码
直播数据传输	直播数据传输协议
	直播数据传输效率
直播数据处理	直播数据处理方式
	直播数据处理效率
直播观看	直播观看默认设置

ii. 各子关注点的候选架构模式

a) 直播源采集方式

✧ 用于区分的参数：

- 易用性，熟练度培训（场景 4）
- 互操作性，需要连接外部设备
- 性能，手机和外部设备直播采集的性能差异

编号	方案名称	易用性	复杂性	视频质量
1	仅支持手机采集	高	低	中
2	仅支持外部录像设备采集	中	中	中、高
3	同时支持手机和外部录像设备采集	高	中	中、高

✧ 选择方案：同时支持手机和外部录像设备采集

✧ 选择原因：客户端有易用性的质量属性需求，手机采集直播源的方式非常简单，而外部录像设备采集则需要用户使用客户端连接外部设备，然后再进行采集，过程比较繁琐、不方便。所以在易用性上来说，应该选择手机采集。而外部录像设备采集的优点是相比于手机采集来说，采集的视频质量高，在视频质量上对用户更加友好。因此，我们选择了同时支持手机和外部录像设备采集的方案，能够综合两者的优点，让用户可以选择使用哪种方式采集，以满足用户的需求。

b) 直播源压缩编码

✧ 用于区分的参数：

- 性能，对客户端压缩的性能要求以及网络传输的容错性（场景 9）
- 易用性，保证在采集过程中用户的操作是正常不卡顿的
- 可伸缩性，用户数量增加（场景 6）

编号	方案名称	压缩率	容错能力	CPU 能力要求	压缩时间
1	MPEG2 压缩标准	低	中	中	短
2	H.264 压缩标准	高	高	高	长

✧ 选择方案：H.264 压缩标准

✧ 选择原因：MPEG 压缩标准属于比较古老的技术，而 H.264 压缩标准是比较新的压缩技术，所以在压缩率和网络传输的容错能力上讲，H.264 压缩率高，文件体积小，视频清晰，优于 MPEG2。但 MPEG2 虽然压缩率低，但对 CPU 的运算能力要求比较低，压缩时间也比较短。但同时考虑到需要进行网络传输，网络传输效率的优先级高于客户端运算的优先级，而通过 H.264 压缩的视频体积小，利于网络传输，同时又有很好的网络容错能力。所以选择 H.264 压缩标准。

c) 直播数据传输协议

✧ 用于区分的参数：

- 性能，客户端进行网络访问（场景 9）
- 可伸缩性，用户数量增加（场景 6）
- 可用性，直播数据流中断（场景 12）

编号	方案名称	可靠性	实时性	连续性
1	TCP 协议	强	弱	坏
2	UDP 协议	弱	强	好

✧ 选择方案：UDP 协议

✧ 选择原因：UDP 协议是无连接的，它本身没有数据校验机制，不会进行数据的校准和分组确认。而 TCP 是面向连接的可靠协议，如果发现分组有丢失现象，会请求进行重传，这样会使后面的分组被阻塞，会导致整个播放过程的延时变大。所以使用 UDP 协议虽然可靠性弱，但是能保证实时性和视频的连续性。同时视频数据的容错能力比较高，即使丢失个别数据也不会对观看效果造成很大影响，所以允许传输不那么可靠。综上，我们选择 UDP 协议进行直播数据的传输。

d) 直播数据传输效率

✧ 用于区分的参数：

- 性能，客户端进行网络访问（场景 9）
- 可伸缩性，用户数量增加（场景 6）
- 可用性，直播数据流中断（场景 12）

编号	方案名称	传输效率	可用性	可容纳用户数量
1	使用分布式服务器	高	高	多
2	提升服务器带宽	高	中	中

3	限制客户端的码率	中低	中	中
---	----------	----	---	---

- ✧ 选择方案：使用分布式服务器，并根据需要提升各服务器带宽
- ✧ 选择原因：直播数据传输效率主要考虑传输速率和传输可用性。传输效率主要由服务器总带宽决定，单一提升某一服务器的带宽的确可以提高传输效率，但使用分布式服务器不仅可以提升总带宽，而且有更高的可用性和容错性。单一服务器的连接并发数是固定的，可容纳用户也有数量上的限制，而使用分布式的多台服务器可以容纳更多的用户。如果某台服务器发生故障，单一服务器就再也不能提供服务，而分布式服务器仍可对用户提供直播数据。而限制客户端的码率并不是长久的方案，首先不满足易用性。所以首选方案是使用分布式服务器，当然也可以根据需求提升分布式服务器的带宽。

e) 直播数据处理方式

- ✧ 用于区分的参数：
 - 可扩展性：需要增加一个新的功能，如回看直播（场景 13）
 - 性能：直播数据处理的性能要求
 - 成本：存储直播数据需要较大成本

编号	方案名称	可扩展性	占用存储	性能要求	成本
1	直接转发	低	无	低	低
2	边存储边转发	高	多	高	高

- ✧ 选择方案：边存储边转发
- ✧ 选择原因：如果选用直接转发而不把直播数据保存在存储服务器中，则可以节省存储空间，对系统的性能要求也没那么高，成本也很低。但直播数据是有用的数据，是用商业价值的，假如后续要求有回看直播的功能，存储直播数据就非常有用，可扩展性很高。使用边存储边转发的方案，既能保证用户能够基本实时观看直播，又能将直播数据存储起来，以供利用。

f) 直播数据处理效率

- ✧ 用于区分的参数：
 - 性能，服务器对于直播数据处理的性能要求
 - 可用性，服务失效需要修复（场景 11）
 - 可用性，直播数据流中断（场景 12）

编号	方案名称	处理速率	可用性	成本
1	存储和转发分离	快	中	中
2	使用多台存储/转发服务器	快	高	高

- ✧ 选择方案：服务器存储和转发模块分离，并且使用多台存储/转发服务器
- ✧ 选择原因：存储和转发分离可以让服务器专心做某一件事情，比如存储服务器只需将数据存储起来，而转发服务器只需将数据转发给各个客户端，这样能够加快数据的处理速率。而使用多台服务器更可以减轻单服务器的负担，更能提高处理速率。如果服务失效

或直播数据流中断，多台服务器仍然可以保证正常的工作状态，用户的直播观看不受影响。

g) 直播观看默认设置

- 用于区分的参数：
- 易用性，正常操作（场景 5）
- 性能，客户端进行网络访问（场景 9）

编号	方案名称	视频质量	视频流畅度
1	质量优先	高	根据网络情况
2	流畅优先	低	高
3	智能推荐	不定	不定

◇ 选择方案：智能推荐

◇ 选择原因：用户在客户端上观看直播的体验很大程度取决于网络情况，如果选择质量优先，则不能保证在网络情况较差时用户能流畅观看直播；如果选择流畅优先，则用户体验变差，视频的清晰度不够。而智能推荐是系统根据当前的网络状况，在质量、流畅度和网络状况中综合考虑，自动选择接收直播视频的码率，这样能达到最好的用户体验。

3.1.3. 第三次迭代

● 步骤一

第三次迭代可以不用此步骤。

● 步骤二

我们选择系统（服务器和客户端）的视频模块进行分解。

● 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 2：操作系统升级	高	中
2	场景 9：客户端进行网络访问，如提出问题、观看直播和视频、数据同步等	高	高
3	场景 10：正常使用客户端	中	低
4	场景 11：服务失效需要修复	高	中

● 步骤四

选择满足架构驱动因素的架构模式。

i. 设计关注点

视频存储：在服务器上存储视频，包括了压缩和存储过程。

视频传输：上传/下载视频的活动。

视频观看：用户在客户端观看视频的活动。

设计关注点	子关注点
视频存储	视频存储方式
视频传输	传输压缩编码
视频观看	视频观看默认设置

ii. 各子关注点的候选架构模式

a) 视频存储方式

✧ 用于区分的参数：

- 性能，服务器响应速度（场景 10）
- 管理复杂度

编号	性能	性能	管理复杂度
1	定时长存储	低	低
2	定空间存储	高	中

✧ 选择方案：定空间存储

✧ 选择原因：定时长存储虽然易于管理，但是非常容易产生磁盘碎片，从而影响服务器响应速度，即使进行磁盘碎片整理，依然效果有限，并且无法阻止碎片继续产生。而定空间存储虽然管理复杂度高，但是能够满足性能的要求，且管理开销在可承受范围内，因此选择定空间存储。

b) 视频传输压缩编码

✧ 用于区分的参数：

- 速度，上传/下载所需时间。
- 质量，即视频画质、是否缺帧等。

编号	方案名称	速度	质量
1	MPEG-4	中	高
2	H.264	高	高

✧ 选择方案：H. 264 压缩标准

✧ 选择原因：MPEG-4 虽然是 MPEG 标准为低比率传输开发的新压缩标准，但是他采用面向对象压缩，因此在传输速度上略慢。而 H. 264 具有高压缩比，高质量图像和良好的网络适应性，可以在较低的带宽上提供高质量图像传输，因此选择 H. 264 编码标准。

c) 直播观看默认设置

✧ 用于区分的参数：

- 易用性，正常操作（场景 5）
- 性能，客户端进行网络访问（场景 9）

编号	方案名称	视频质量	视频流畅度
----	------	------	-------

1	质量优先	高	根据网络情况
2	流畅优先	低	高
3	智能推荐	不定	不定

✧ 选择方案：智能推荐

✧ 选择原因：用户观看视频时适合哪种质量的视频取决于所处环境的网络速度，所以默认为只能推荐模式，之后用户可以自行更改。

3.1.4. 第四次迭代

● 步骤一

第四次迭代可以不用此步骤。

● 步骤二

此次迭代选择系统中的可穿戴设备连接模块进行分解。可穿戴设备连接模块是系统健康管理这一核心功能点的重要子模块，它主要分为可穿戴设备认证与连接和可穿戴设备数据同步两个功能点。对该模块进行分解，对进一步设计该系统的体系结构有重要影响。

● 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 1：手环升级或需要连接新设备	中	中
8	场景 8：经过授权的访问	高	中
10	场景 10：正常使用客户端	中	低
14	场景 14：需要与可穿戴设备进行交互	高	中

● 步骤四

选择满足架构驱动因素的架构模式。

i. 设计关注点

设计关注点	子关注点
可穿戴设备认证	客户端与设备认证方式
可穿戴设备数据同步	数据同步方式
可穿戴设备支持	可穿戴设备支持

ii. 各子关注点的候选架构模式

a) 可穿戴设备认证

✧ 用于区分的参数

- 安全性：保证客户端与可穿戴设备实现数据传输的过程将涉及的用户的敏感性私密数据的安全

编号	方案名	安全性	复杂度
1	用户-设备认证	中	低
2	客户端-设备认证	中	中
3	用户-客户端-设备认证	高	中

✧ 选择方案：用户-客户端-设备认证

✧ 选择原因：用户-设备验证与设备-设备验证都只在用户与可穿戴设备间或移动客户端与可穿戴设备间进行两方验证。考虑到系统涉及到用户、客户端设备与可穿戴设备三方，且客户端与可穿戴设备实现数据传输获取的都是用户的敏感性私密数据，出于对用户隐私的保护，系统对数据安全性要求非常高，因此为了防止被数据侦听或破解，造成对用户隐私的侵犯，选择安全性最高的用户-设备-设备验证方案。

b) 可穿戴设备数据同步

✧ 用于区分的参数：

- 性能：保证用户使用的流畅性（场景 10）
- 易用性：保证良好的用户体验，能及时同步数据（场景 10，场景 14）

编号	方案名	易用性	资源占用率
1	仅采用系统轮询	中	中
2	仅支持由用户自定义数据同步	中	低
3	同时支持系统轮询与用户自定义数据同步	高	中

✧ 选择方案：同时支持系统轮询与用户自定义数据同步

✧ 选择原因：同时支持系统轮询与用户自定义数据同步方案既保证了初始时系统已有默认设置完成数据同步任务，又给用户提供了根据个人习惯自定义设置的接口，能实现更好的用户体验。

c) 可穿戴设备支持

✧ 用于区分的参数：

- 可维护性：系统要求具备能扩展支持新的可穿戴设备类型及其数据处理（场景 1）

编号	方案名	代码修改程度	复杂度
1	通过配置文件配置各类可穿戴设备	低	高
2	通过修改代码封装各类可穿戴设备	高	中

✧ 选择方案：通过修改代码封装各类可穿戴设备

✧ 选择原因：当前可穿戴设备的种类繁多，但是业内并无统一的规范和标准，各家厂商所提供的 API 和数据格式都不同，企图通过配置文件灵活的升级对设备的支持这一方案在当前还是无法实现的，因此只能选择通过修改代码来满足这一要求。

3.1.5. 第五次迭代

● 步骤一

第五次迭代可以不用此步骤。

● 步骤二

此次迭代选择系统中的数据库接口模块进行分解。

● 步骤三

确定架构驱动因素，如下表所示：

#	架构驱动因素	重要性	难易度
1	场景 2：操作系统升级	高	中
2	场景 7：未经授权的访问	高	高
3	场景 9：客户端进行网络访问，如提出问题、观看直播和视频、数据同步等	高	高
4	场景 15：服务器或客户端移植到其他环境	中	中

● 步骤四

选择满足架构驱动因素的架构模式。

i. 设计关注点

设计关注点	次级关注点
数据存储	数据存储方式
	数据存储架构
	数据库结构
数据传输	数据传输方式
	数据传输过程监测
数据加密	数据加密方式

ii. 各子关注点的候选架构模式

a) 数据存储方式

✧ 用于区分的参数

- 可移植性：在数据需要迁移时保证迁移工作的简便（场景 15）
- 可维护性：保证数据存储兼容（场景 2）
- 安全性：保证数据存储不被篡改（场景 7）

编号	方案名	读写速度	操作容易度	安全性
1	在线存储	快	容易	中
2	近线存储	慢	中等	中
3	脱机存储	极快	困难	低
4	异地保护	块	中等	高

✧ 选择方案：在线存储

✧ 选择原因：要求保证数据的可移植性和数据的访问速度，近线存储的数据访问速度较慢，

脱机存储需要人为存储，操作过于繁琐，异站存储降低了数据的可移植性，故选择对访问速度及安全性及可移植性都有保障的在线存储方式。

b) 数据存储架构

✧ 用于区分的参数

- 可移植性：在数据需要迁移时保证迁移工作的简便（场景 15）
- 可维护性：保证数据存储兼容（场景 2）
- 安全性：保证数据存储不被篡改（场景 7）

编号	方案名	数据量	访问速度
1	嵌入式架构	小	中
2	X86 架构	中	慢
3	云技术	高	快

✧ 选择方案：云存储

✧ 选择原因：系统要求极大的数据存储量以及数据访问的速度、直播时视频数据的稳定和清晰度，嵌入式架构和基于 X86 的架构在数据存储量上无法满足项目要求，故选择云存储架构。

c) 数据库结构

✧ 用于区分的参数

- 可移植性：在数据需要迁移时保证迁移工作的简便（场景 15）
- 可维护性：保证数据存储兼容（场景 2）
- 性能：保证直播和视频数据稳定（场景 9）

编号	方案名	访问速度	语句复杂度
1	网状数据库	快	高
2	关系数据库	中	低
3	树状数据库	中	高
4	面向对象数据库	快	高

✧ 选择方案：关系数据库

✧ 选择原因：场景 5 要求系统保证数据访问的速度及简明的数据结构。树形数据库和网状数据库结构较复杂，面向对象数据库访问速度较慢，故选择二者能兼顾的关系数据库。

d) 数据传输方式

✧ 用于区分的参数

- 安全性：保证数据传输中的安全（场景 7，12）
- 性能：保证数据传输的迅速（场景 5，9）

编号	方案名	方向	同步/异步
1	单工传输	固定方向	同步
2	半双工传输	两个方向	同步

3	全双工传输	两个方向	异步
---	-------	------	----

✧ 选择方案：全双工传输

✧ 选择原因：场景 12,9 要求系统能保证直播和视频数据流的稳定，并且与用户的交互异步进行，故选择能两个方向异步传输的全双工传输方式。

e) 数据传输过程监控

✧ 用于区分的参数

- 安全性：对不正常的数据传输进行警报（场景 7）
- 性能：保证数据传输分流进行提升传输速度（场景 5，9）

编号	方案名	占用带宽	检测覆盖率
1	传输全程监控	高	>90%
2	传输抽样监控	低	>50%

✧ 选择方案：数据传输全程监控

✧ 选择原因：场景 7 要求系统有极高的安全性，需要识别恶意传输的数据流，同时兼顾系统性能。虽然全程监控开销比抽样大，但是能够保证用户数据在传输过程中的安全性，安全性更加重要。故选择全程监控的方案。

f) 数据加密方式

✧ 用于区分的参数

- 安全性：保证加密的复杂度，最大限度的防止破解（场景 7）
- 性能：加密和解密不能影响系统的正常响应（场景 5）

编号	方案名	复杂度	对系统响应的影响
1	对称加密	高	低
2	非对称加密	很高	低

✧ 选择方案：对称加密

✧ 选择原因：场景 7 要求系统保证数据的安全性，兼顾系统性能，能及时响应客户需求。非对称加密与对称加密都能保证系统数据很好的安全性，但是非对称加密对于客户端和服务端的负担更大，为了保证系统的性能，故选择对称加密方案。

3.2. ATAM 评估过程

3.2.1. 质量属性效用树

质量属性	属性求精	场景
性能	直播响应时间	客户端向服务器/客户端发出一次请求后的响应时间应少于 2s (M, M)
		客户端请求直播和视频数据的缓冲时间应不大于 10s (M, M)

	吞吐量	在系统正常运行的状态下，系统可以保证同时响应 10000 名用户的操作 (H, H)
可用性		系统服务 24 小时可用 (H, M)
		直播数据流中断时，数据流源修复时间<10 分钟 (M, M)
可伸缩性		在系统正常运行状态下，可支持的并发用户数量不低于 10000 (H, M)
		用户请求直播连接的平均请求时间不超过 5 秒 (H, M)
安全性	机密性	系统验证用户身份，拒绝不在用户权限范围内的服务和数据访问，并通知用户 (H, H)
	完整性	系统拒绝未经授权的访问 (H, H)
可维护性		当可穿戴设备升级时，维护相应代码的时间应不超过 3 小时，并且受影响的代码量小于 2% (H, M)
		当操作系统的新特性影响到系统原有功能的实现时，系统应该能在 0.5 个人月的成本下完成更新，并且成本不应超过总开发成本的 5% (H, M)
		当增加新的建议文件格式时，维护相应代码的时间应不超过 2 小时，并且受影响的代码量小于 1% (H, M)
可移植性		服务器或客户端移植到其他环境，完成移植所花费的代价不得高于 4 人/月 (H, H)
互操作性	数据接口	本系统可与指定可穿戴设备进行数据传输，交互成功率大于 95% (H, H)
易用性	正常操作	用户成功执行一项细节操作，如上传可穿戴设备数据、查看历史统计数据、观看健身课程直播等操作，完成单项操作的时间不超过 2 分钟 (H, H)
	熟练度培训	使用该系统的直播拍摄相关工作人员，掌握直播功能操作所需时间在一周以内 (H, M)

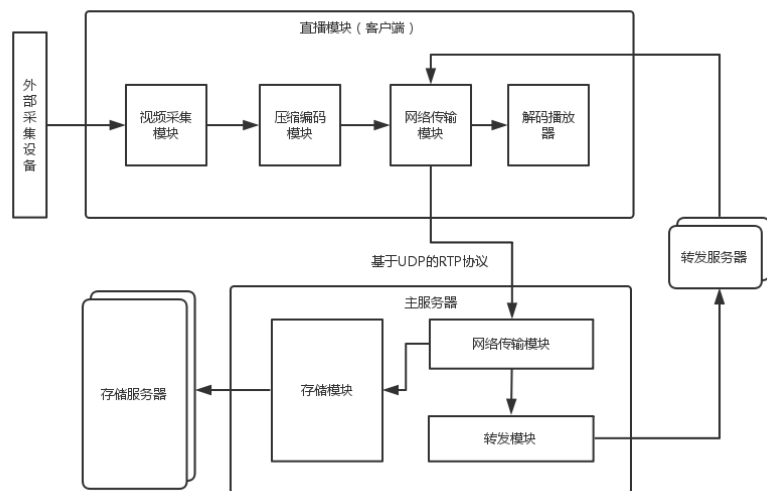
3.2.2. 架构方法分析

场景号：A1	用户请求直播后获得响应			
属性	性能			
环境	正常操作			
刺激	用户请求直播			
响应	客户端向服务器/客户端发出一次请求后的响应时间应少于 2s			
构架决策	敏感点	权衡点	有风险决策	无风险决策
使用分布式服务器	S1	T1, T2		N1
提升服务器带宽	S2	T1		N2
边存储边转发	S3			N3

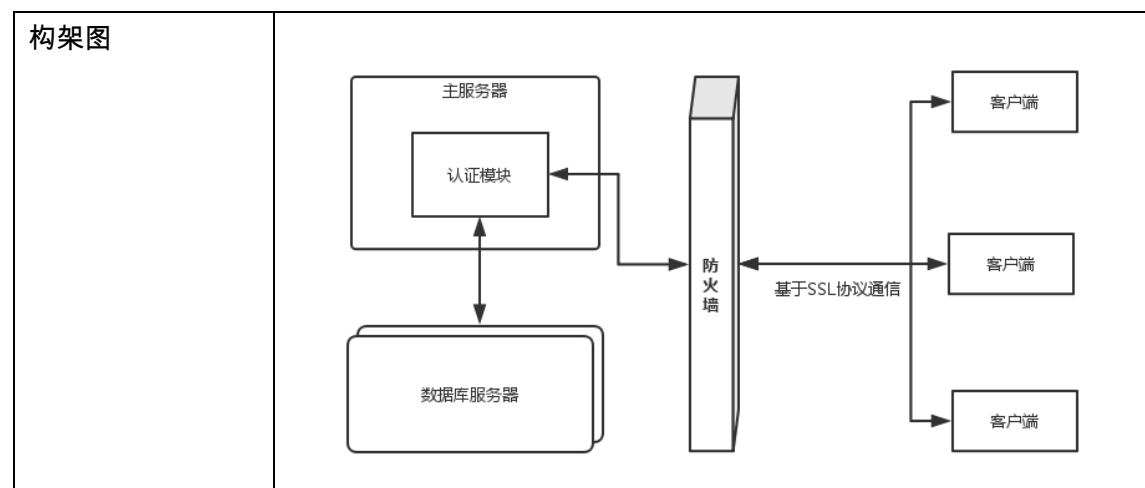
存储和转发分离并 使用多台存储/转 发服务器	S4	T1,T2		N4
理由	<p>使用分布式服务器和多台存储/转发服务器提高了系统的性能但同时 也极大地增加了成本开销，而且维护多台服务器的管理更复杂，降低 了系统的可维护性。</p> <p>提升服务带宽将增加资源成本。</p>			
构架图	<pre> graph LR subgraph Client ["直播模块 (客户端)"] direction LR VCM[视频采集模块] --> CEM[压缩编码模块] CEM --> NTM1[网络传输模块] NTM1 --> DPL[解码播放器] end subgraph MainServer ["主服务器"] direction TB SSM[存储服务器] <--> SM[存储模块] SM --> NTM2[网络传输模块] NTM2 --> FM[转发模块] end subgraph ForwardingServer ["转发服务器"] direction TB FS[转发服务器] end NTM1 -- "基于UDP的RTP协议" --> NTM2 FM --> FS </pre>			

场景号：A2	用户请求直播和视频数据后缓冲			
属性	性能			
环境	正常操作			
刺激	用户请求直播和视频数据			
响应	客户端请求直播和视频数据的缓冲时间应不大于 10s			
构架决策	敏感点	权衡点	有风险决策	无风险决策
使用分布式服务器	S1	T1, T2		N1
提升服务器带宽	S2	T1		N2
存储和转发分离并 使用多台存储/转 发服务器	S4	T1,T2		N4
理由	<p>客户端直播和视频数据的缓冲时间是和服务器带宽以及用户自身网络 带宽相关的。使用分布式服务器和提升服务器带宽可以提升服务器的 总带宽，从而提升直播和数据数据的传输速率，减少缓冲时间，但是这 也会增加服务器的成本开销，而且维护多台服务器也变得复杂。</p>			

构架图



场景号：A3	未经授权的访问			
属性	安全性			
环境	系统正常运行			
刺激	用户访问不在权限范围内的数据			
响应	系统验证用户身份，拒绝用户的访问			
构架决策	敏感点	权衡点	有风险决策	无风险决策
防火墙	S5	T3	R1	
双因素身份验证	S6	T4	R2	
SSL	S7	T5	R3	
理由	<p>采用防火墙技术提高了系统的安全性但防火墙的并发连接数限制容易导致拥塞或溢出，对系统的可用性造成威胁；另外由于无法全面控制网络、应用程序和数据流，防火墙缺乏完整的基于会话级别的监控能力，很难预防新的未知攻击。</p> <p>在重要的操作执行前进行双因素身份验证可以提高对非法访问的抵御能力，当额外增加的安全操作降低了系统的易用性。</p> <p>SSL 的加密解密功能需要占用计算处理资源，完成大量的计算任务将会影响性能。</p>			



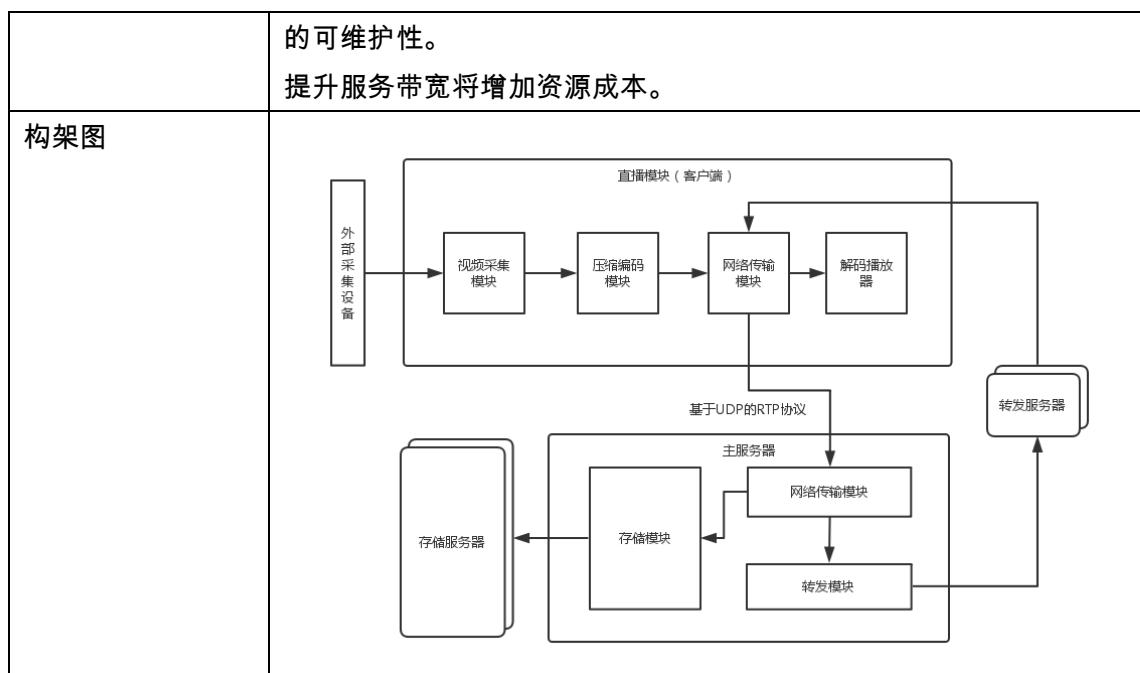
场景号：A4	当可穿戴设备升级时，维护相应代码的时间应不超过 3 小时，并且受影响的代码量小于 2%			
属性	可维护性，可扩展性			
环境	系统正常运行			
刺激	可穿戴设备要升级			
响应	运维人员升级系统并向客户端发布更新通知			
构架决策	敏感点	权衡点	有风险决策	无风险决策
模块化设计	S8			N5
源码修改扩展功能	S9	T6	R4	
理由	<p>采用模块化架构设计将系统的功能分块设计，则可将扩展功能所需修改的部分限制在模块内部，实现低耦合高内聚，减少当系统维护时涉及到的代码量。</p> <p>当前可穿戴设备的种类繁多，但是业内并无统一的规范和标准，各家厂商所提供的 API 和数据格式都不同，企图通过配置文件灵活的升级对设备的支持这一方案在当前还是无法实现的，因此只能选择通过修改代码来满足这一要求，但直接修改源码在测试安全性，保证性能和系统集成度方面将带来负面影响。</p>			

场景	服务器或客户端移植到其他环境，完成移植所花费的代价不得高于 4 人/月			
属性	可移植性			
环境	系统部署			
刺激	服务器或客户端要移植到其他环境			
响应	开发人员完成和测试移植并将系统部署到新的环境			
构架决策	敏感点	权衡点	有风险决策	无风险决策
采用原生语言开发 Android(java),IOS(s	S10	T6	R5	

wift)				
服务器采用 Java 开发	S11			N6
理由	<p>本系统对性能要求很高，因此采用原生语言开发有利于提高系统性能，但由于要分别开发 Android 和 IOS 两套系统，在时间、人力、资金等方面都要投入将近两倍的成本，而且系统的移植性也将受到限制。</p> <p>Java 语言在 JVM 上运行，跨平台性好且安全性较高。</p>			

场景号：A5	直播数据流中断恢复			
属性	可用性			
环境	系统正常运行			
刺激	直播数据流中断			
响应	系统数据流源修复时间<10 分钟			
构架决策	敏感点	权衡点	有风险决策	无风险决策
推流专用服务器	S12	T7		N7
备用数据源	S13	T8		N8
提升服务器带宽	S14	T8		N9
理由	<p>架设推流专用的服务器，可以使得直播的中断便于定位和恢复，提高了系统的可用性。但是可能会增加系统的成本和负担，降低系统的性能。</p> <p>使用备用数据源可以在直播中断时最快速度的恢复直播信息，利于维护人员的修复工作，提高了系统的可用性。但是备用数据源的存在增大了系统负担，在大多数情况下直播正常进行时是对系统资源的极大浪费，降低了系统的性能。</p> <p>提升服务器带宽有利于减少直播因为推流速度不够而引起的中断现象，提高了系统的可用性，但是增大了系统的资源开销。</p>			

场景号：A6	10000 个用户同时请求服务			
属性	可伸缩性			
环境	正常操作			
刺激	用户请求服务			
响应	服务器响应率达到 99%			
构架决策	敏感点	权衡点	有风险决策	无风险决策
使用分布式服务器	S1	T1, T2		N1
提升服务器带宽	S2	T1		N2
使用分布式数据库	S12	T1, T2		N10
理由	<p>使用分布式服务器和分布式数据库提高了系统的性能但同时也极大地增加了成本开销，而且分布式的布局使得维护更为困难，降低了系统</p>			



3.2.3. 敏感点列表

编号	描述
S1	分布式服务器决定了服务器端的架构
S2	提高服务器带宽对系统的性能有重要影响
S3	边存储变转发，既能保证用户能够基本实时观看直播，又能将直播数据存储起来，保证了良好的用户体验
S4	存储和转发分离并使用多台存储/转发服务器的决策将带来较大的开发成本
S5	防火墙技术用于保证系统网络访问安全，提高了系统安全性
S6	双因素身份验证主要用于对重要操作的权限控制和再次确认，提高对非法访问的抵御能力
S7	SSL 加密是系统的重要安全决策，用于保护用户的隐私数据等
S8	模块化设计决策对系统的可扩展性、可修改性、可维护性等质量属性都有重要影响。
S9	通过直接修改源码扩展功能对系统的职责合理分离和封装要求很高，因此需要慎重考虑
S10	客户端开发语言的选择对系统的性能、可维护性、可移植性以及开发成本都有很大的影响
S11	服务器开发语言的选择对服务器端的安全性、可移植性以及开发成本都有重要影响
S12	专用推流服务器保证了直播的大部分时间可用，对可用性有重大影响
S13	备用数据源保证系统最大时间内可用，提高了系统的可用性
S14	带宽的提升保证了直播的正常进行，减少了直播中断的现象，提高了可用性

S12	分布式数据库对系统结构有重要影响，同时也影响着所有涉及数据传输的功能。
-----	-------------------------------------

3.2.4. 权衡点列表

编号	描述
T1	提高了系统的性能但同时也极大地增加了成本开销
T2	提高了系统的性能，降低了系统的可维护性
T3	提高了安全性，但对可用性造成了威胁
T4	提高对非法访问的抵御能力，当额外增加的安全操作降低了系统的易用性
T5	提高了安全性，当可能降低系统性能
T6	提高了系统的性能，但也提高了开发成本和维护成本，降低了可维护性
T7	提高了可用性，但是降低了系统的性能，增大系统负担
T8	提高了系统的可用性，但是增大了系统的成本和资源开销

3.2.5. 有风险决策列表

编号	描述
R1	防火墙无法全面控制网络、应用程序和数据流，防火墙缺乏完整的基于会话级别的监控能力，很难预防新的未知攻击
R2	额外增加的安全操作降低了系统的易用性，可能导致用户流失
R3	SSL 的加密解密功能需要占用计算处理资源，完成大量的计算任务将会影响性能
R4	直接修改源码可能在不断的更改之中对系统的完整性、安全性可测试性造成负面影响，导致系统复杂度不可控
R5	原生语言开发客户端极大的提高了成本且导致系统的跨平台性和可移植性不好

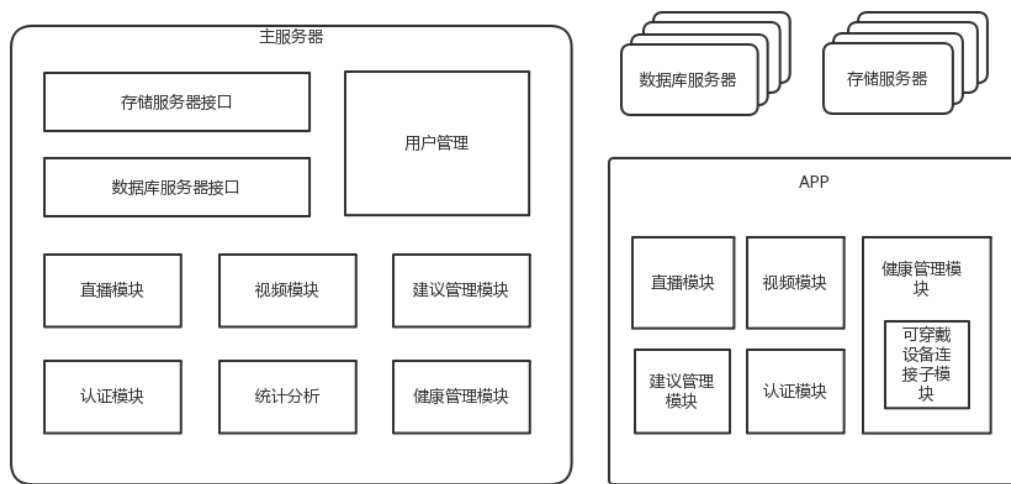
3.2.6. 无风险决策列表

编号	描述
N1	分布式服务器是服务器开发的趋势，技术已经成熟且对性能有较大提升
N2	提高服务器带宽将会带来更好的性能和用户体验
N3	使用边存储边转发的方案，既能保证用户能够基本实时观看直播，又能将直播数据存储起来，以供利用
N4	存储与转发服务器分离使得各服务器专注于单一职责，降低系统耦合度且利于分流和合理配置服务器资源
N5	模块化是在传统设计基础上发展起来的一种新的设计思想，现已成为一种新技术被广泛应用，降低了软件的复杂度，有助于提高软件的可维护性、可扩展性

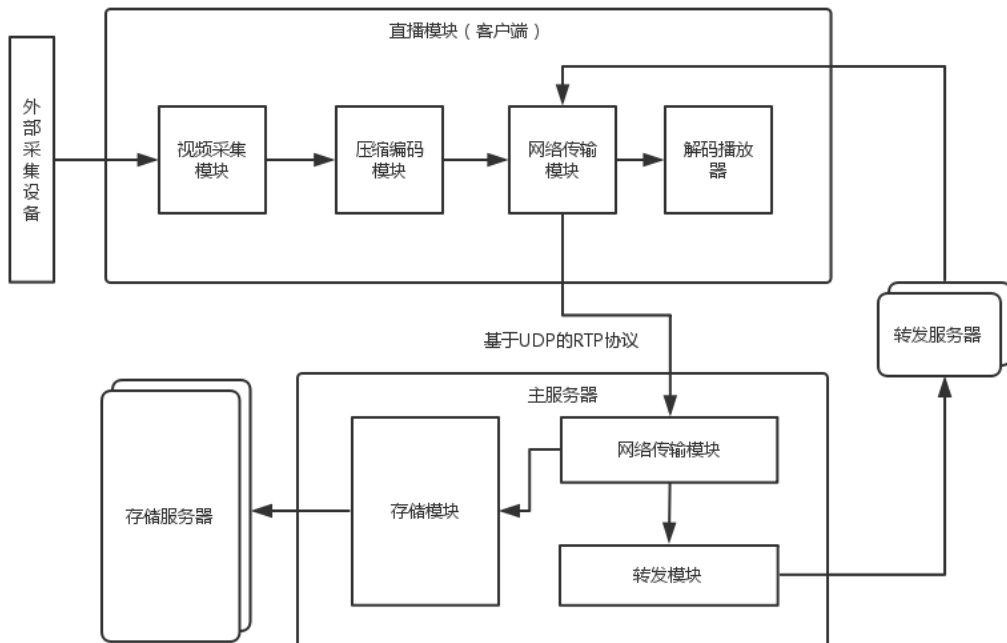
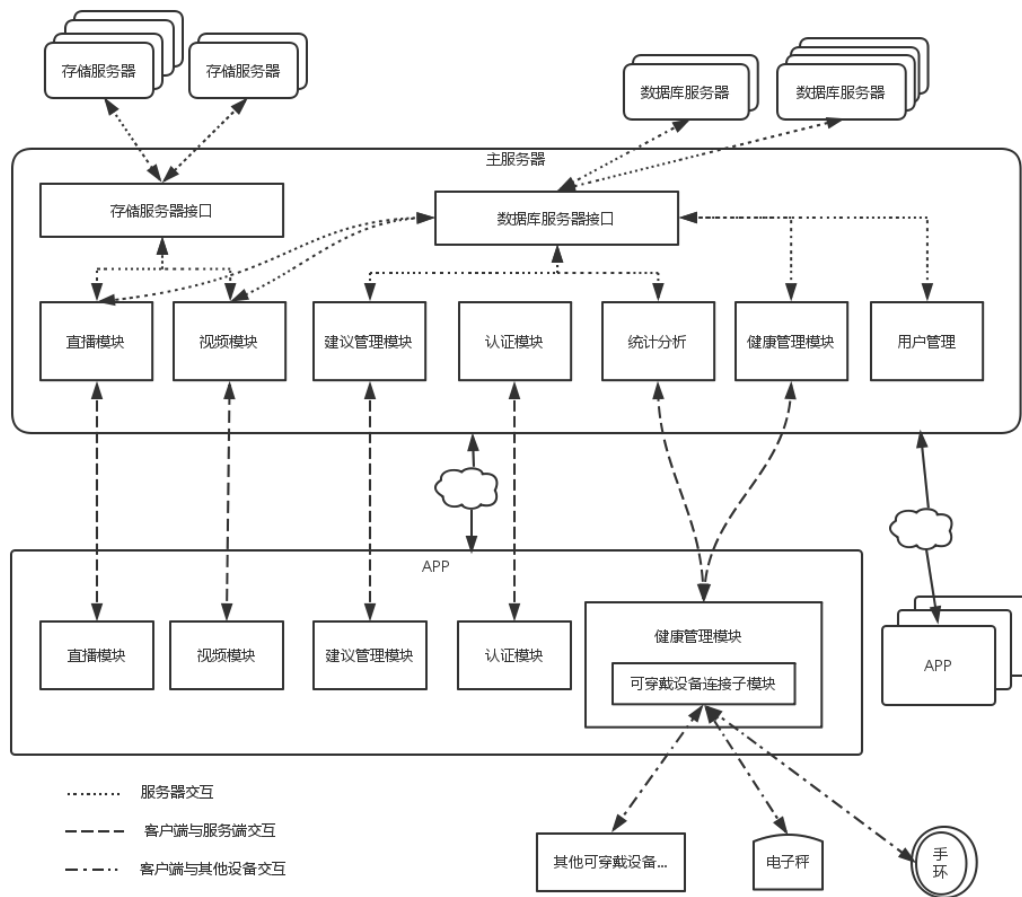
	等
N6	Java 适合于开发大型的应用系统，系统易维护、可复用性、可移植性好
N7	专用推流服务器保证了直播最大限度的可用性
N8	备用数据源保证了直播的流畅度和易用性，也提高了系统可用性
N9	提升带宽可以保证推流的稳定性，保证直播的流畅和易用性，提高了系统的可用性
N10	分布式数据库可以很好的满足可伸缩性需要

3.3. 架构图

1) 模块视图



2) C&C 视图



4. P2P 与 C/S 架构对比

4.1. 性能对比

P2P	C/S
<ul style="list-style-type: none">● 节点越多，网络性能越好，网络随着规模增大而越发稳固● P2P 能充分利用 Peer 的资源，提高传输效率和通信质量	<ul style="list-style-type: none">● 服务器的存储能力和处理能力以及所在网络的吞吐量是该模式性能的瓶颈，随着节点增加，服务器负载越来越重，一旦服务器崩溃，整个网络也随之瘫痪● 需要通过增加服务器数量来提升性能，成本高

4.2. 安全性对比

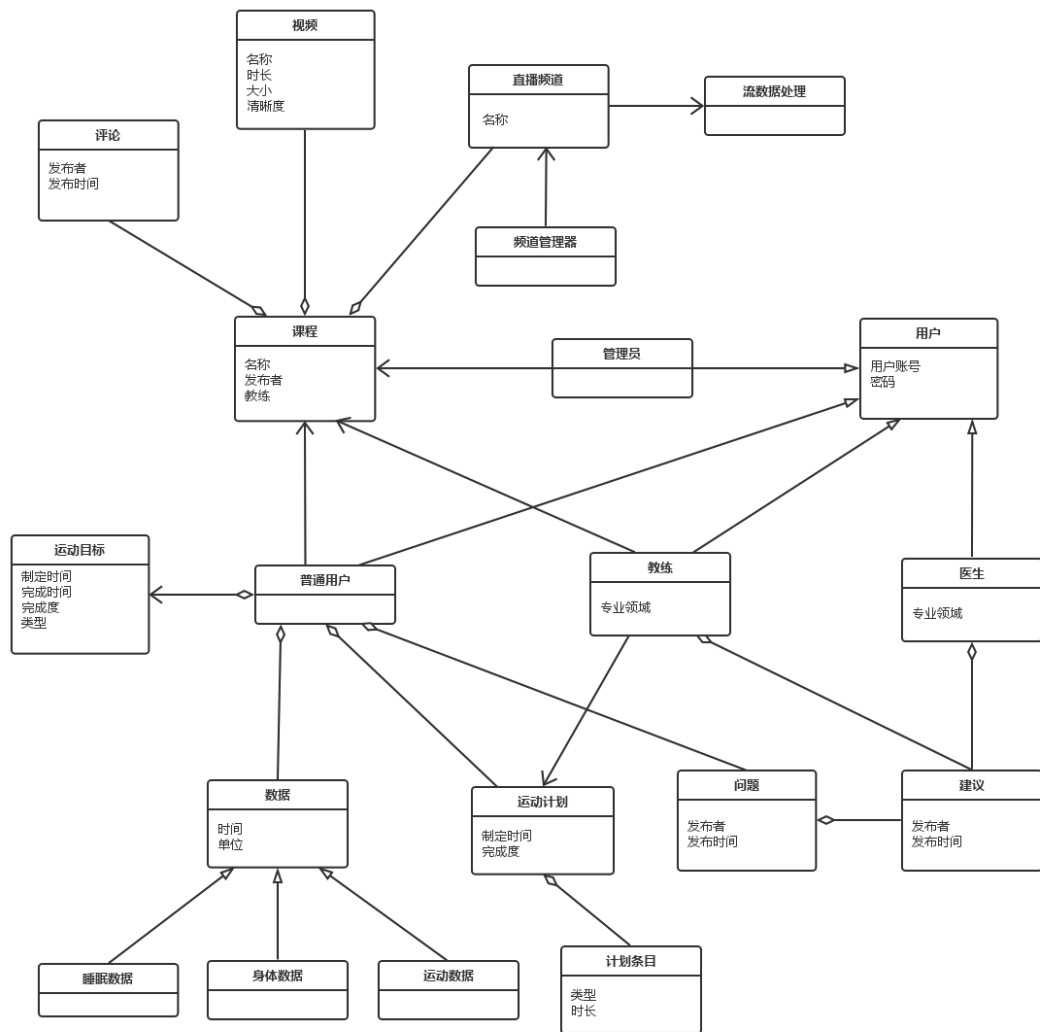
P2P	C/S
<ul style="list-style-type: none">● 信息共享，用户自由度高，带来随机性和不确定性，数据安全性难以保证● 安全策略、备份策略复杂，成本高	<ul style="list-style-type: none">● 数据与控制流方向单一，高度集中，安全措施实现相对简单● 一旦服务器被攻击，损失很大

4.3. 资源利用对比

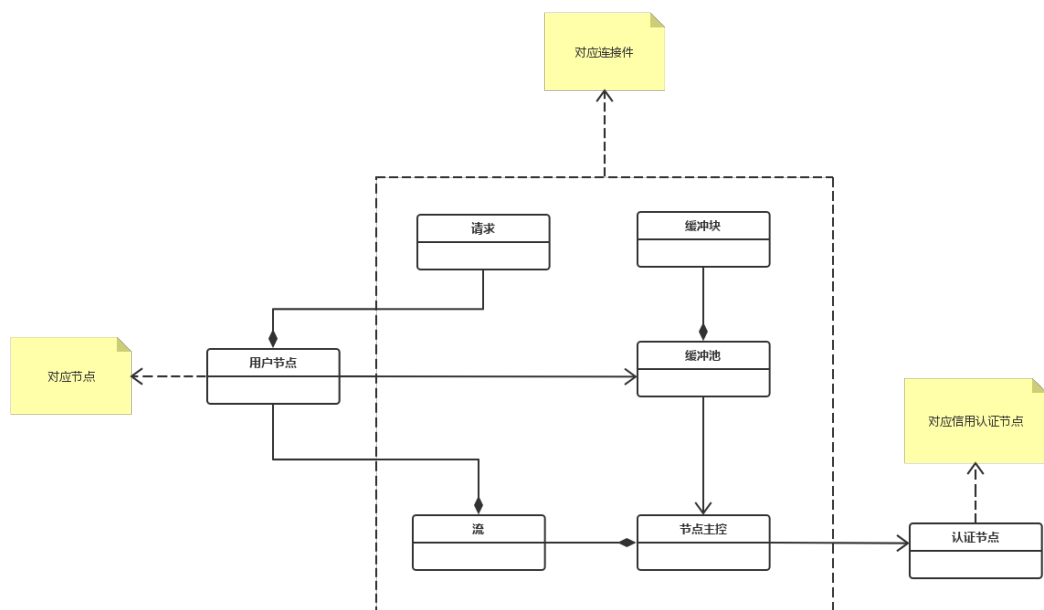
P2P	C/S
<ul style="list-style-type: none">● 能够有效的利用所有节点上的各种资源● 占用客户端存储、处理资源	<ul style="list-style-type: none">● 大量有一定计算与存储能力的客户端的资源被闲置● 相同数据的分发造成服务器和网络带宽的浪费● 资源更新成本高

5. 系统类图设计

经过 P2P 架构和 C/S 架构的设计、评估、对比过程，最终决定系统采用 P2P 架构，并设计类图如下所示：



类与组件、连接件的映射关系如下图所示：



6. 挑战和经验

此次项目最大的挑战来自于对于 P2P 架构以及直播过程的不熟悉，为了顺利完成架构设计，小组成员查阅了大量资料，在直播部分进行了深入的了解，对直播系统有了更深的了解。

通过此次项目实践，小组成员对 ADD 方法和 ATAM 评估方面有了更加全面的认识。ADD 是以质量为主要矛盾，功能为次要矛盾来进行体系结构设计，重点在于质量属性上。ATAM 是评价软件构架的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况，而且可以使我们更清楚地认识到质量目标之间的联系，如何权衡诸多质量目标。

同时我们对架构设计的流程也更加熟悉，但也了解到一个好的架构的设计的过程是困难和复杂的，既要进行架构设计又要进行评估，才能确保架构设计的质量。在进行架构设计和评估时，还要注意多考虑本系统的特性和侧重点，对重要的部分要多花些功夫。作为体系架构师，需要对软件开发的各方面都要十分清楚了解，包括各种决策，解决方案可能的影响等等，需要不断地汲取知识和积累很多的经验才能得到更好的成果。

此外，这次团队的人数比较多，因此如何高效地利用人力资源进行合作成为了另一个侧重点。在每次团队会议中，每位成员都会参与讨论，涉及的话题包括：系统特色功能、两种 c&c 架构的选择、ADD 方法、ATAM 方法的汇总和基于此的比较等。我们会提前准备议题，尽可能地保证每次团队会议都围绕着解决问题的思路开展。另外在分工上，我们把两个基本 c&c 架构划分给两个小组（4 人）分别使用 ADD、ATAM 方法进行体系结构设计和评估，为的是保证大家尽可能多的考虑到不同架构的特点，便于之后对两个架构详尽比较，再进行选择。由此，我们整个团队合作的过程都相对比较顺利。

7. 组员和分工

组员	学号	分工
王静漪	131250116	参与质量属性划分，负责易用性场景的描述。基于 ADD、ATAM 方法进行对 P2P 架构的体系结构设计与评估：参与视频播放模块性能这一质量属性的 ADD 过程讨论；负责用户管理的安全性这一部分的架构设计。代表小组进行演讲，展示了系统功能、团队设计过程和基于此对两种架构模式的比较。
唐珊珊	131250128	负责安全性相关场景的编写；参与 P2P 架构设计的 ADD 和 ATAM 过程，负责互操作性部分；参与直播模块的架构决策讨论。
周睿	131250153	参与模块划分，参与架构设计的讨论，负责对数据库接口模块的 ADD 过程和 ATAM 过程。
王望月	131250173	参与讨论了小组产品——Mover 的功能，选取 C/S 和 P2P 两种 C&C 风格的架构来实现。参与讨论了 C/S 架构的 ADD

		和 ATAM，完成了 Mover 主要功能模块——视频模块的 ADD，ATAM 的可伸缩性的场景。最后完成了展示用 ppt 的 C/S 部分。
廖丹琪	131250182	与小组成员讨论 C/S 架构的总体结构和模块划分，负责架构相关设计图的绘制和可穿戴数据模块的 ADD 分析和相关模块的 ATAM 评估。
蔡晓莹	131250183	负责性能相关场景的编写；参与 P2P 架构设计的 ADD 和 ATAM 过程，负责可用性部分；参与直播模块的架构决策讨论；负责最终文档汇总与排版。
许悠	131250184	参与小组需求讨论，参与了 P2P 架构设计的 ADD、ATAM 过程；完成 ADD 中健康管理模块可维护性的分析；参与 ADD 中直播模块的分析；完成可维护性的 ATAM 评估；完成可维护性相关 PPT 部分。
王振聪	131250218	参与小组需求讨论，参与了 C/S 架构设计的 ADD、ATAM 过程；完成 ADD 中直播模块的设计以及 ATAM 中性能的决策评估；协助王静漪完成 presentation，展示 C/S 架构的 ADD 和 ATAM 过程；完成最终文档中 C/S 架构部分的汇总。