

## 流媒体系统总体设计 Theia-0.1

版本修订记录				
编号	日期	版本	修订人	修 订 内 容
1	2019.7.14	0.1	张涛	初始版本，总体设计

### 目标

- 1, 验证流媒体相关技术的可行性
- 2, 探索关键技术最佳实现
- 3, 探索关键指标性能
- 4, 验证流媒体服务的业务流程
- 5, 探索 AI 技术与多媒体技术相结合的产品形式
- 6, 为 AI 和多媒体技术提供落地支持

### 功能

1. 主播功能（需登录）：
  - 客户端发起个人直播
  - PC 网页发起个人直播
  - Obs 进行个人直播
  - 开启连线功能
  - 批准连线请求
  - 断开连线
  - 设置连线时画面布局（大小窗口，对等窗口等）
  - 是否开启人脸检测
2. 观众功能；
  - 查看房间列表
  - 进入直播房间
  - 申请连线（需登录）
  - 留言（需登录）
3. 客户端支持
  - PC 浏览器：Chrome, Firefox
  - PC 客户端：windows

## 业务定义：

1. 用户
  - 通过 web 客户端和 PC 客户端使用本系统的人，称为用户
  - 用户可以查看房间列表
  - 用户可以观看直播
  - 登录后的用户可以进行直播和连线
2. 直播 (Live)
  - 用户发起的一个直播叫做 Live
3. 直播间 (Room)
  - 一个用户有自己的直播间
  - 观众可以通过直播间，观看直播
  - 观众可以在直播间，发出连线请求
4. 连线
  - 观众可以在正在直播的直播间内发出连线请求
  - 房主同意后，进行连线
  - 直播间的其他观众，将看到两人共同直播

## 技术定义：

1. RtpStream：一个 Live 的所捕获的所有音视频媒体信息经过 mpeg-ts 封装后构成的 RTP 流称为一个 RtpStream，可以包含多路音频和多路视频；
2. RtpChannel：一个端到端的单向 Rtp 数据传输通道；
  - 一个 RtpChannel 上可以传递一个或者多个 RtpStream；
  - 由客户端向服务端推送数据的 Rtp 通道，定义为：上行 RtpChannel；
  - 由客户端从服务端拉取数据的 Rtp 通道，定义为：下行 RtpChannel；
  - 一个 Live 的数据传输，包含一个上行 RtpChannel，若有连线则还有一个下行 RtpChannel；
3. Rtp 流的头里，包含了 Rtp 的 Live 标识和 Room 标识；

Figure 1 媒体数据流向图

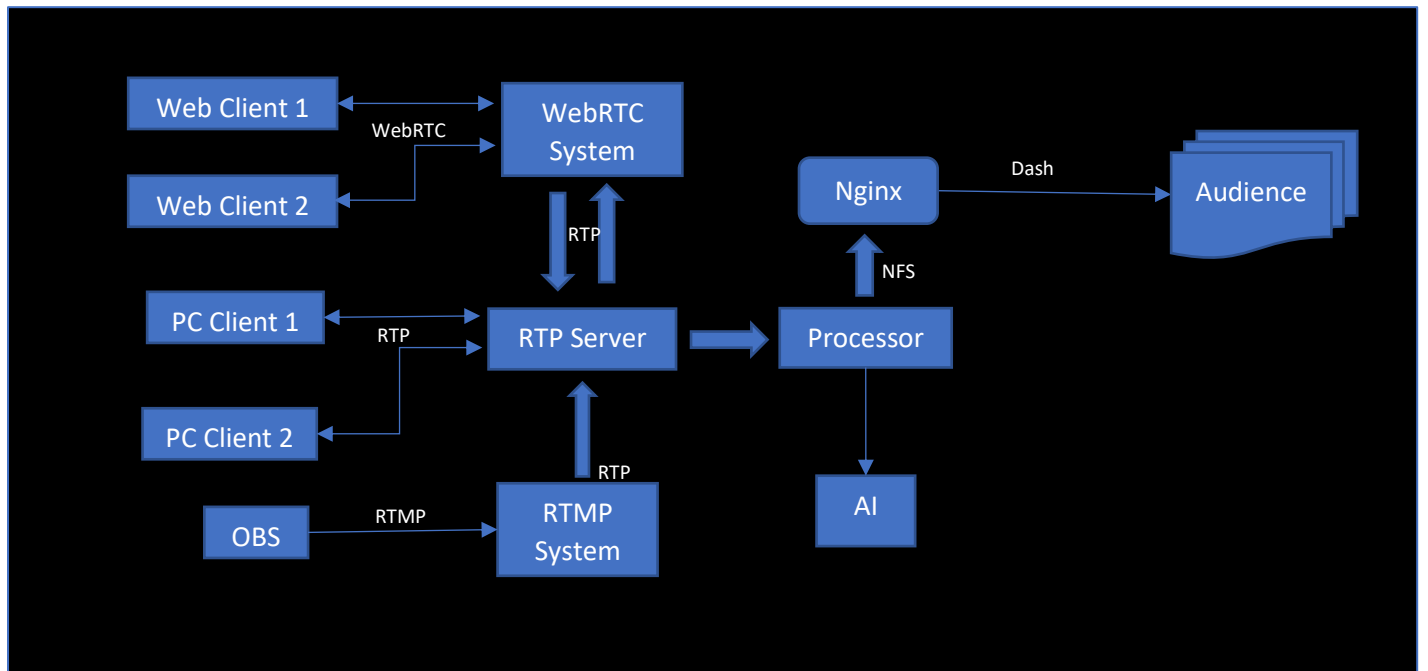
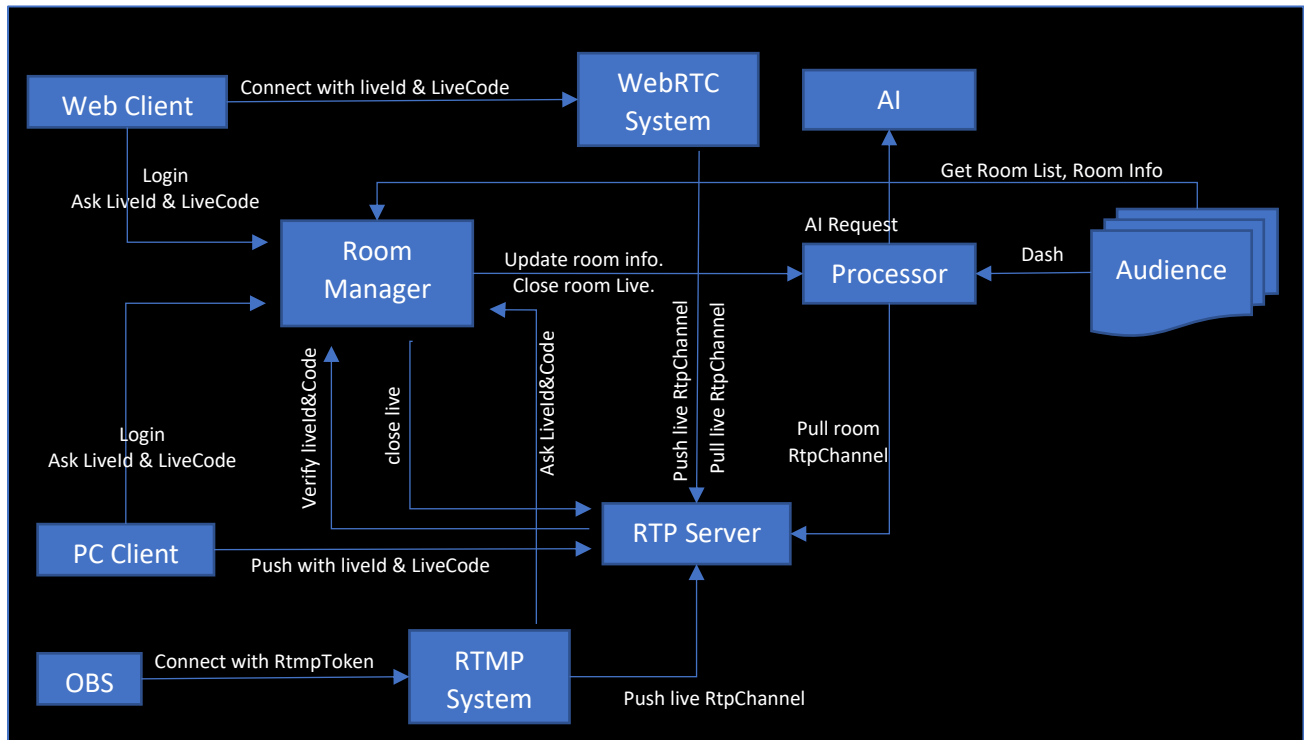


Figure 2 业务调用关系图



## Room Manager 主要 API:

- 用户登录
  - 输入: userName, userPassword
  - 则返回用户信息和 Token;
- 建立 websocket 连接
  - 输入: UserId, Token
  - 建立 websocket 连接
- 申请直播 (主播调用, 通过 websocket)
  - 输入: UserId, Token
  - 查询该用户的 RoomId, 并对应生成一对 LiveId 和 LiveCode
  - 将 LiveId 与 UserId, RoomId 进行对应
  - 返回: LiveId, LiveCode
- 校验直播推流是否授权 (流服务器调用, 暂时不做鉴权)
  - 输入: LiveId, LiveCode
  - 验证 LiveId 与 LiveCode 是否有效
  - 如果有效, 通知 Processor 房间信息变更
  - 返回 true 或 false
- 客户端申请连线 (观众调用, 通过 websocket)
  - 输入: UserId, RoomId
  - 若当前房间状态可以连线
    - 向房主 WebSocket 发送连线请求
  - 若当前房间状态无法连线 (例如房主关闭连线功能)
    - 返回消息提示
- 房主拒绝连线 (通过 websocket)
  - 输入: RoomId, 申请者 UserId
  - 给申请者的 WebSocket 发拒绝消息, 并断开 WebSocket
  - 返回状态给房主
- 房主同意连线 (通过 websocket)
  - 输入: RoomId, 申请者 UserId
  - 为连线者生成一对 LiveId 和 LiveCode
  - 将 LiveId 与 UserId, RoomId 进行对应
  - 给申请者的 WebSocket 发送生成的 LiveId, LiveCode, 以及房主的 LiveId
  - 通知 Processor 更新房间信息
  - 并通过房主的 WebSocket 发送连线者的 LiveId
  - 返回连线者的 UserId 和 LiveId 给房主
- 房主断开连线 (通过 websocket)
  - 输入: RoomId
  - 通知 Processor 更新房间信息
  - 通过 WebSocket 通知连线者断开

- 断开与连线者的 WebSocket
- 获取房间列表
  - 输入：无
  - 返回房间列表信息
- 观众查询房间
  - 输入：RoomId
  - 通过 RoomId 向 Processor 查询 mpd 信息
  - 返回房间信息，以及 mpd 地址
- RtmpToken 换取 LiveId 和 LiveCode（RTMP System 调用）
  - 输入：RtmpToken
  - 若 RtmpToken 有效，则生成 LiveId 和 LiveCode 对
  - 将 LiveId 与 UserId，RoomId 进行对应
  - 返回 LiveId，LiveCode

#### Processor 主要 API:

- 更新房间信息
  - 输入：RoomId，LiveId List，布局信息
  - 返回状态
- 查询 mpd 地址
  - 输入：RoomId
  - 返回 mpd 地址
- 关闭房间直播
  - 输入：RoomId

#### RTP Server 主要 API

- 断开直播推流
  - 输入：LiveId
  - 返回状态
- 申请推流（通过 UDP）（方案 1）
  - 输入：LiveId，LiveCode
  - 返回 ssrc，port
  - 然后客户端使用 ssrc 和 port 开始 RTP 推流
- 申请推流（通过 post）（方案 2）
  - 输入：LiveId，LiveCode
  - 返回 ssrc，port
  - 然后客户端使用 ssrc 和 port 开始 RTP 推流
- 拉流（通过 UDP）
  - 输入：liveId List

- 向申请端发送指定流的 RTP 包

Rtmp System 主要流程:

- 接收推流
  - Obs 使用 RTMP 向 RTMP Server 推流
  - 获取连接参数中的 RtmpToken
  - 使用 RtmpToken 换取 LiveId 和 LiveCode
  - 允许 obs 推流
  - 并将 rtmp 流转为 rtp 流推入 rtmp server
- rtmp 流转为 rtp 流推入 rtmp server
  - 通过 RtmpToken 从 rtmp server 进行拉流
  - 将 rtmp 流进行解封装, 并重新封装为 mpeg-ts 包, 发往 RtpServer
  - 若 n 分钟没有 rtmp 流没有新的数据, 就结束该转发流程

PC 客户端主要流程:

- 登录 (http post)
  - 向 RoomManager 发送 userName 和 password
  - 获取返回的用户信息和 token
  - 将 token 妥善保存
- 建立 websocket
  - 为了实现与 RoomManager 实时交互, 需要建立 websocket
  - 输入: userId 和 Token
  - websocket 连接建立
- PC 客户端直播推流
  - 向 RoomManager 获取 LiveId 和 LiveCode
  - 用申请到的 LiveId 和 LiveCode 向 RTP Server 或者 WebRTC System 进行连接
  - 开始推流
- PC 客户端向 PC 客户端房主进行连线
  - 向 RoomManager 申请 LiveId, LiveCode, 和房主的 LiveId
  - 用申请到的 LiveId 和 LiveCode 向 RTP Server 或者 WebRTC System 进行连接
  - 开始推流
  - 并使用房主的 LiveId 拉流
- PC 客户端向 web 客户端房主进行连线
  - TODO
- 房主接受连线
  - 同意连线

- 获得连线者的 UserId, LiveId
- 使用连线者的 LiveId 进行拉流

WebRTC 客户端主要流程:

- 登录 (http post)
  - 向 RoomManager 发送 userName 和 password
  - 获取返回的用户信息和 token
  - 将 token 妥善保存
- 建立 websocket
  - 为了实现与 RoomManager 实时交互, 需要建立 websocket
  - 输入: userId 和 Token
  - websocket 连接建立
- Web 客户发起直播
  - Web 客户端通过 UserId 和 Token 向 Room Manager 获取 LiveId 和 LiveCode
  - 将 LiveId 和 LiveCode 发送 WebRTC System
  - WebRTC System 通过 LiveId 和 LiveCode 完成 RtpServer 的鉴权
  - WebRTC System 获取用户的 webRTC 流, 并将其转发到 RtpServer
- Web 客户端向 WebRTC 房主进行连线
  - Web 客户端通过 UserId, RoomId 和 Token 向 Room Manager 获取 LiveId, LiveCode, 以及房主的 LiveId
  - 将自己的 LiveId, LiveCode, 以及主播的 LiveId 发送 WebRTC System
  - WebRTC 为房主和连线者之间, 构建 WebRTC 通讯
  - WebRTC System 获取连线者的 webRTC 流, 并将其转发到 RtpServer
- Web 客户端向 PC 客户端房主进行连线
  - TODO

RTP 推流鉴权流程 (方案 1) :

- 1, PC client 首先向 Room Manager 申请 LiveId 和 LiveCode
- 2, PC client 通过 UDP 发送 LiveId 和 LiveCode 给 RTP Server
- 3, RTP Server 拿到 LiveId 和 LiveCode 向 RoomManager 验证有效性
- 4, 验证通过后 RTP Server 分配 ssrc, 并将 ssrc 和 LiveId 关联
- 5, RTP Server 将 ssrc 通过 UDP 回复给 PC client
- 6, PC Client 开始使用 ssrc 通过 RTP 向 RTP Server 进行推流

RTP 推流鉴权流程（方案 2）：

- 1, PC client 首先向 Room Manager 申请 LiveId 和 LiveCode
- 2, PC client 通过 post request 发送 LiveId 和 LiveCode 给 RTP Server
- 3, RTP Server 拿到 LiveId 和 LiveCode 向 RoomManager 验证有效性
- 4, 验证通过后 RTP Server 分配 ssrc, 并将该 ssrc 与 LiveId 进行关联
- 5, RTP Server 通过 post response 将 rtp 端口和分配好的 ssrc 回复给 PC client
- 6, PC Client 开始通过获得得端口和 ssrc 向 RTP Server 进行 RTP 推流

拉流暂不做鉴权