

## 流媒体系统总体设计 Theia-0.2.0

版本修订记录				
编号	日期	版本	修订人	修 订 内 容
1	2019.7.14	0.1.0	张涛	初始版本，总体设计
2	2019.7.16	0.1.1	张涛	WebRTC System 流程细化，Web 客户端连线流程细化，PC 客户端连线流程细化
3	2019.8.23	0.2.0	张涛	根据第一阶段的经验，对内容进行整体性调整，针对能力和业务相分离进行了模块划分

### 目标

- 1, 能力、业务相分离，达到更好的模块间复用
- 2, 降低子系统间的耦合
- 3, 完善账户体系以及业务流程
- 4, 增强系统稳定性
- 5, CV 的尝试

### 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）
  - CV 特效
  - 账户系统管理
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、点赞（需登录）
3. 客户端支持
  - 观众端：Chrome, Firefox, Safari （PC and 手机）
  - 连线客户端：windows, Mac
  - 特效客户端：windows, Mac

#### 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

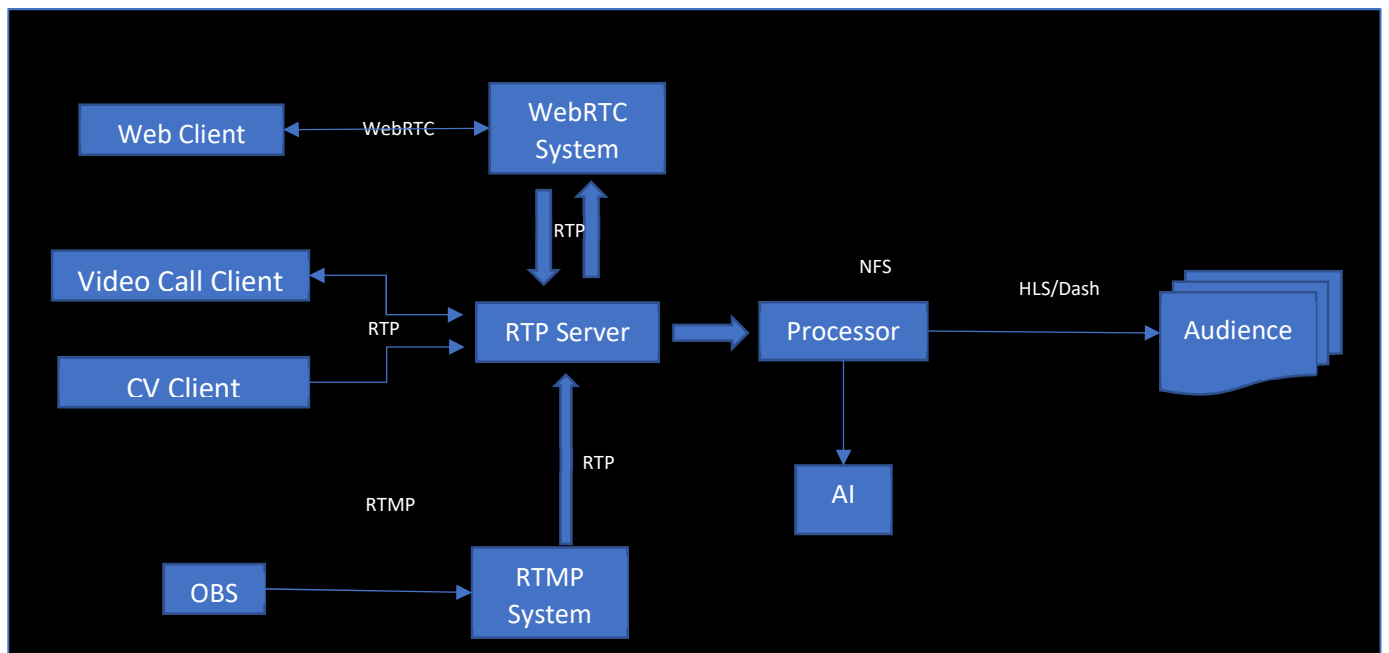
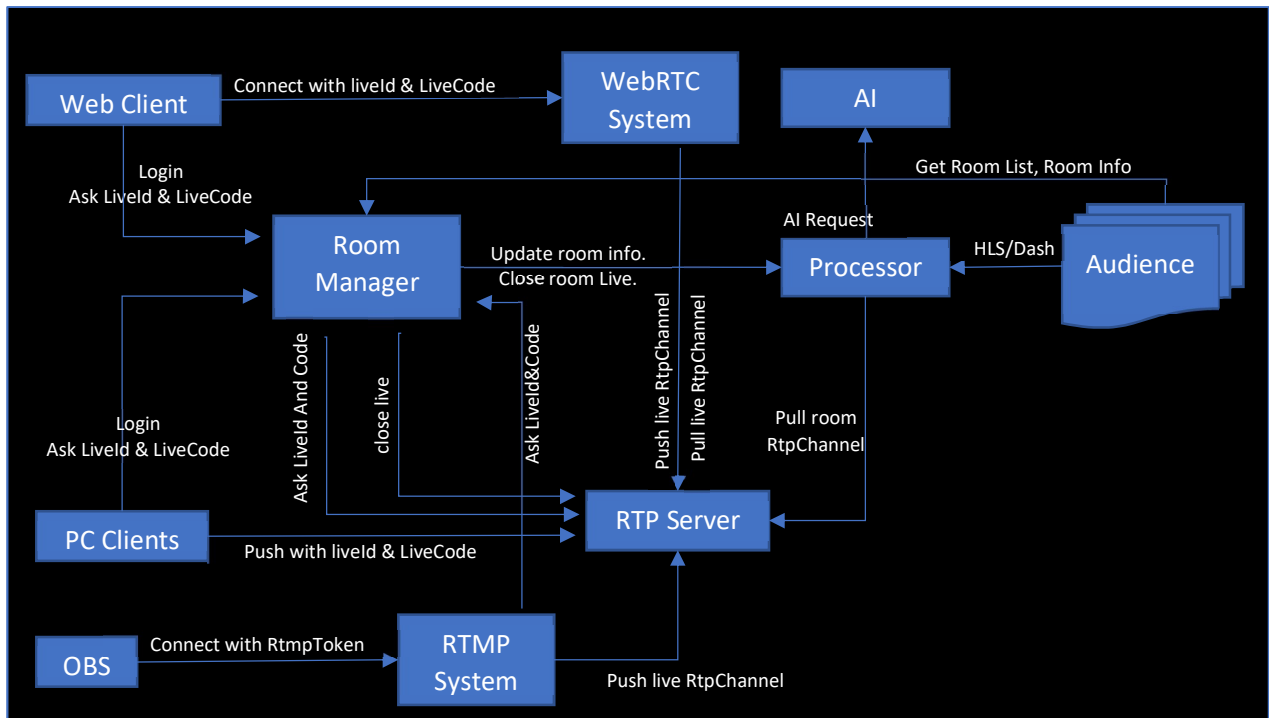


Figure 2 业务调用关系图



## 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, 并向 RtpServer 申请一对 LiveId 和 LiveCode
  - 将 LiveId 与 UserId, RoomId 进行对应
  - 返回: LiveId, LiveCode
- 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
  - 为连线者向 RtpServer 申请一对 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, 则向 RtpServer 申请一对 LiveId 和 LiveCode
  - 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

WebRTC System 主要流程：（推后）

- Web 客户端直播
  - 与 web 客户端建立 webRTC 连接
  - 将媒体流采用 mpeg-ts 封装转发 RtpServer
- Web 客户端向 web 客户端房主连线
  - 与 web 客户端连线者建立 webRTC 连接
  - 根据连线者 LiveId 和房主 LiveId，为两者建立 WebRTC 通讯
  - 将连线者的媒体流采用 mpeg-ts 封装转发 RtpServer
- Web 客户端向 PC 客户端房主连线
  - 与 web 客户端连线者建立 webRTC 连接
  - 通过房主 LiveId 确认房主是 PC 客户端
  - 通过 WebRTC 获取 web 客户端多媒体流
  - 通过 web 客户端连线者 LiveId 向 RtpServer 进行 mpeg-ts 推流
  - 通过 PC 客户端房主 LiveId 向 RtpServer 进行拉流，并解 mpeg-ts
  - 通过 WebRTC 向 web 客户端连线者发送多媒体流
- PC 客户端向 Web 客户端房主连线
  - Web 客户端房主告知 WebRTC System 连线者的 LiveId
  - 通过连线者 LiveId 确认连线者是 PC 客户端
  - 通过 PC 客户端连线者 LiveId 向 RtpServer 进行拉流，并解 mpeg-ts
  - 通过 WebRTC 向 web 客户端房主发送多媒体流

连线客户端主要流程：

- 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 客户端向房主进行连线
  - PC 客户端向 RoomManager 获取 LiveId，LiveCode，和房主的 LiveId
  - 用申请到的 LiveId 和 LiveCode 向 RTP Server 推流
  - 并使用房主的 LiveId 拉流
- PC 客户端房主被连线
  - 同意连线
  - 通过 Room Manager 获得连线者 UserId，LiveId
  - 使用连线者的 LiveId 拉流

#### CV 客户端主要流程

- 登录（与连线客户端一致）
- 建立（与连线客户端一致）
- 直播推流（与连线客户端一致）

#### WebRTC 客户端主要流程（推后）：

- 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，开始 WebRTC 推流
- Web 客户端向房主进行连线
  - Web 客户端通过 UserId，RoomId 和 Token 向 Room Manager 获取 LiveId，LiveCode，以及房主的 LiveId

- Web 客户端将自己的 LiveId, LiveCode, 以及主播的 LiveId 发送给 WebRTC System 进行连接
- Web 客户端开始 WebRTC 推流和拉流
- Web 客户端房主被连线
  - 同意连线
  - 将连线者的 LiveId 发送给 WebRTC System
  - 通过 WebRTC System 拉流

RTP 推流鉴权流程:

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

拉流暂不做鉴权

模块划分:

- 1, 直播间服务器
- 2, 连线客户端
- 3, CV 客户端
- 4, 播放引擎
- 5, 采集引擎
- 6, 观众 Web 端 (PC, 手机)
- 7, RTMP 转换服务
- 8, 流媒体服务器与 SDK(RTP MCU, 流鉴权)
- 9, 媒体处理服务器 (画面融合, 声音融合)
- 10, 媒体流分发服务

模块主要功能:

直播间服务器



- 1, 账号体系维护
- 2, 用户鉴权
- 3, 观看、直播、连线等业务控制
- 4, 留言、互动

#### 连线客户端:

- 1, 直播推流
- 2, 观众观看直播
- 3, 多观众连线
- 4, 留言互动

#### CV 客户端

- 1, 直播推流
- 2, CV 处理
- 3, 留言互动

#### 播放引擎

- 1, 输入为二进制 mpegts 流 (inputStream)
- 2, 支持输出图像和音频数据
- 3, 提供 Javafx canvas, 能够自行同步的绘制画面和播放声音
- 4, 支持仅播放声音, 支持仅播放画面
- 5, 同步/缓冲/分辨率/帧率/图像的调整
- 6, 录制功能

#### 采集引擎

- 1, 列出视频采集设备
- 2, 列出音频采集设备
- 3, 可以输出采集到的图像和音频数据
- 4, 指定视频和音频采集设备, 输出二进制 mpegts 流
- 5, 支持仅采集声音, 支持仅采集图像
- 6, 同步/缓冲/分辨率/帧率/图像/码率的调整
- 7, 录制功能

#### 观众 Web 端 (PC, 手机)

- 1, 注册、登录、个人信息等账号功能
- 2, 展示直播间信息列表
- 3, 直播间观看直播
- 4, 直播间进行留言、互动

#### RTMP 转换服务

- 1, rtmp 鉴权
- 2, 接收 rtmp 码流
- 3, 分辨率/帧率/图像/码率的调整
- 4, 推送到流媒体服务器

流媒体服务器与 SDK(RTP MCU, 流鉴权)

- 1, LiveId 与 LiveCode 维护
- 2, 媒体流鉴权
- 3, 接收媒体流
- 4, 分发媒体流/流组

媒体处理服务器 (画面融合, 声音融合)

- 1, 直播间的情况配置对画面进行拼接
- 2, 多画面分辨率/帧率/图像/码率的调整
- 3, 音频的处理与融合
- 4, 将处理后的媒体流推到分发服务器

媒体流分发服务

- 1, 支持 Dash 分发
- 2, 支持 HLS 分发
- 3, 支持 WebSocket 分发