

WebRTC一站式直播解决方案

阿武

现就职于YY直播，曾就职华为、中兴

擅长领域：C++/golang后端研发

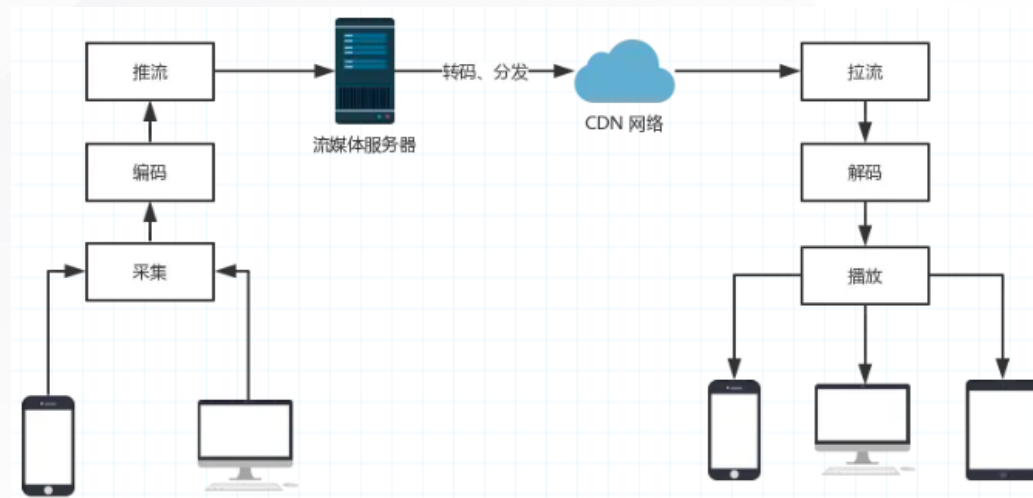
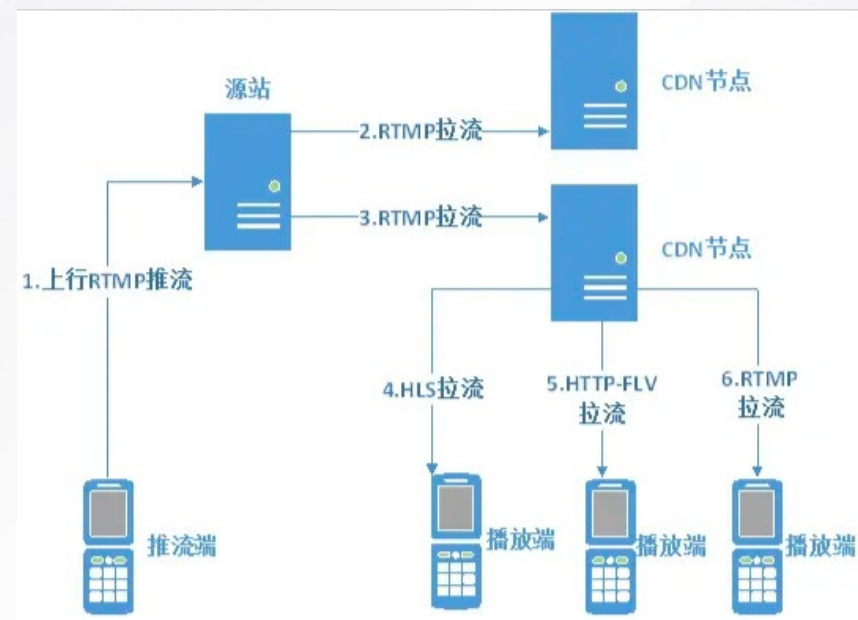
1. webrtc, ffmpeg直播技术
2. nginx, bfe, service-mesh基础架构

WebRTC一站式直播解决方案

- 1. 通用直播模型
- 2. 直播产品的种类
- 3. RTC应用场景
- 4. RTC-SFU解决方案
- 5. WebRTC直播系统架构
- 6. WebRTC直播系统demo演示

通用直播模型

直播技术难与易之争



直播产品的种类

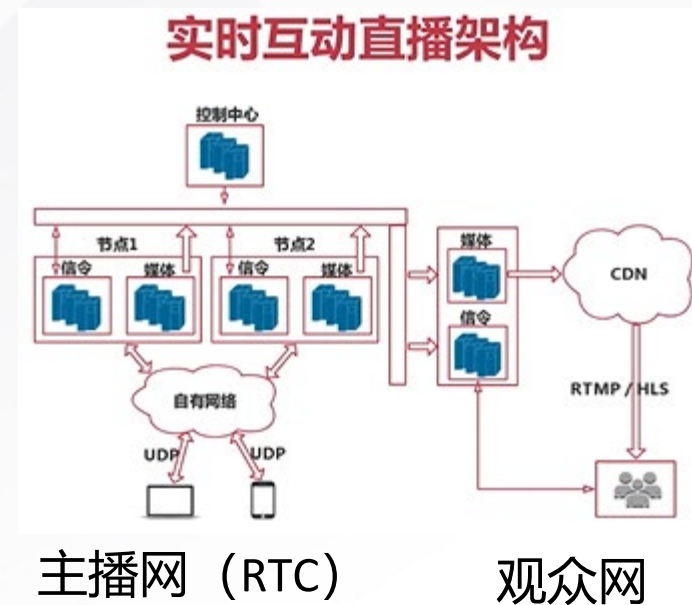
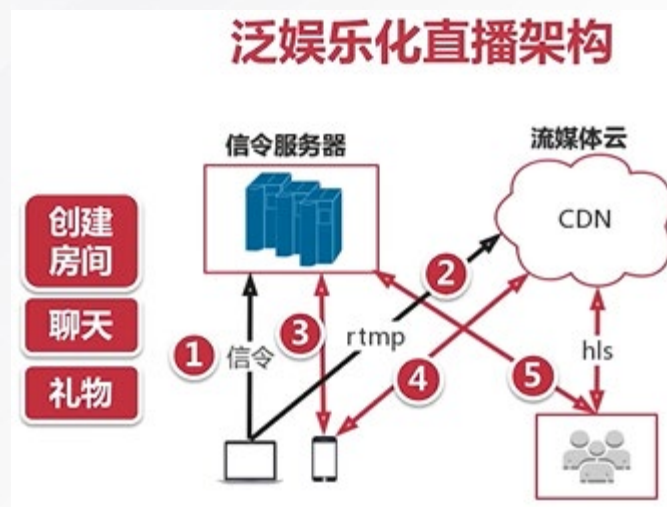
直播产品的种类

- 泛娱乐化直播

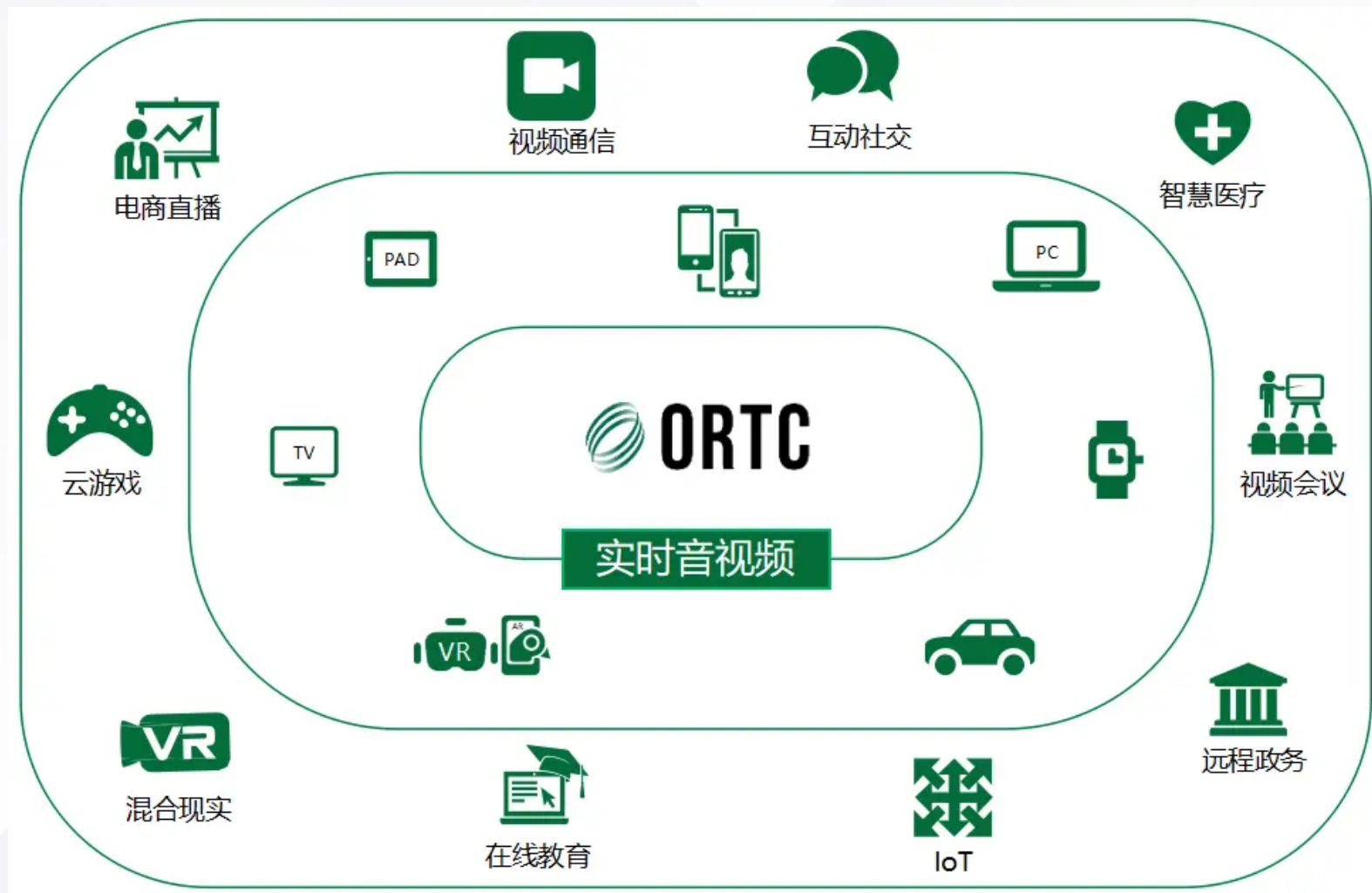
抖音、YY、快手等娱乐直播，虎牙、斗鱼等游戏直播

- 实质互动直播

音视频会议、教育直播等，如声网、zoom













RTC应用场景



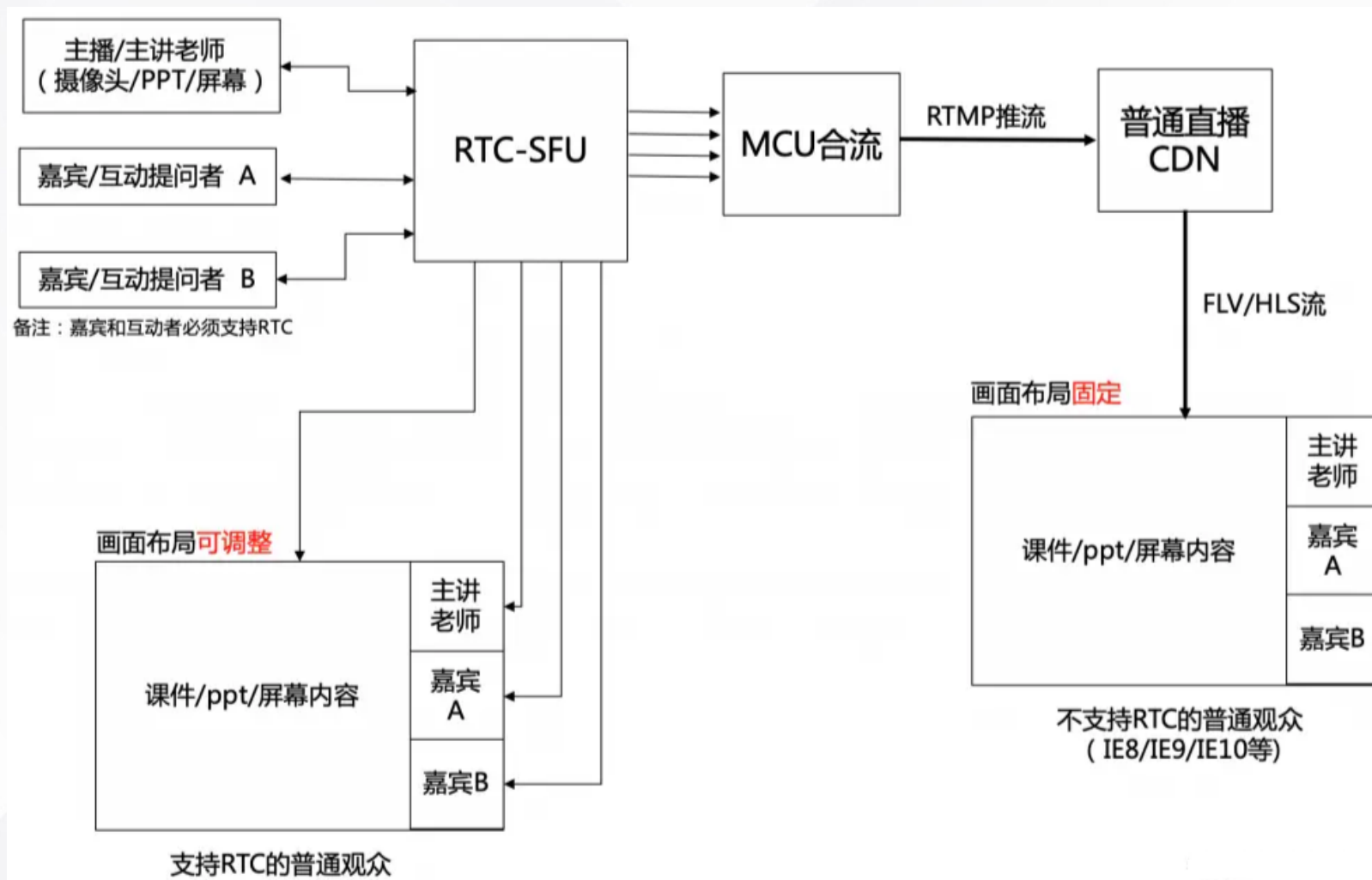
RTC-SFU解决方案

| Project | SFU | MCU | License |
|-----------|-----|-----|---------|
| Janus | ✓ | ✓ | GPL v3 |
| Licode | ✓ | ✓ | MIT |
| MediaSoup | ✓ | ✗ | ISC |
| Kurento | ✓ | ✓ | Apache2 |
| Jitsi | ✓ | ✗ | Apache2 |

1. 视频会议
2. 在线教育 (小班课)
3. 客服系统

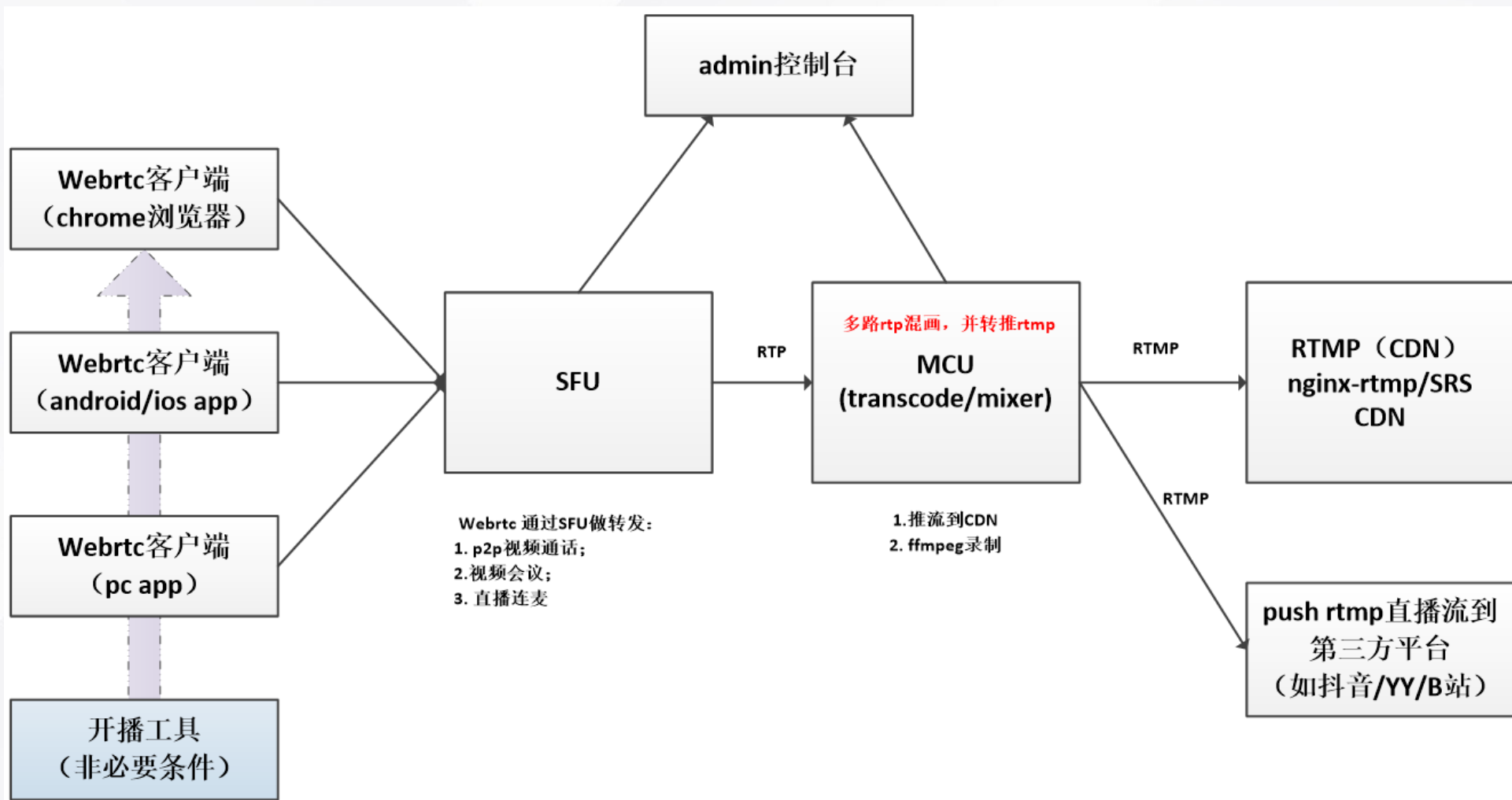
| | | | | |
|--|---|---|--|--|
|   Use at your own peril |   Use for specific projects when other frameworks don't cut it |   Use when your UI/UX is like Jitsi Meet (or close enough to it) |   Use for scalable conferencing and streaming |   Use when you like Go or need specialized integrations |
|--|---|---|--|--|

RTC直播系统架构

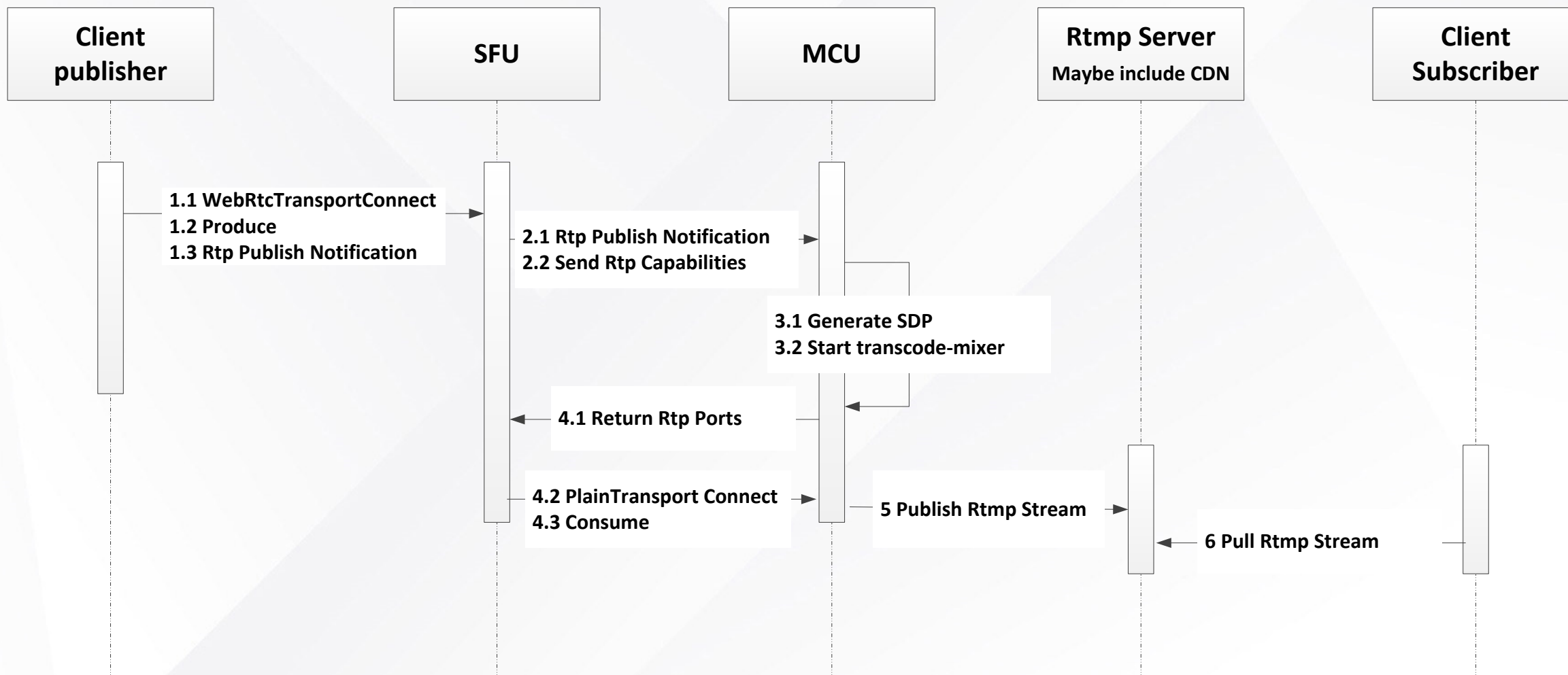


RTC直播系统架构

- sfu音视频流转发服务
- mcu转码、混流服务
- admin直播控制台
- 成本最低

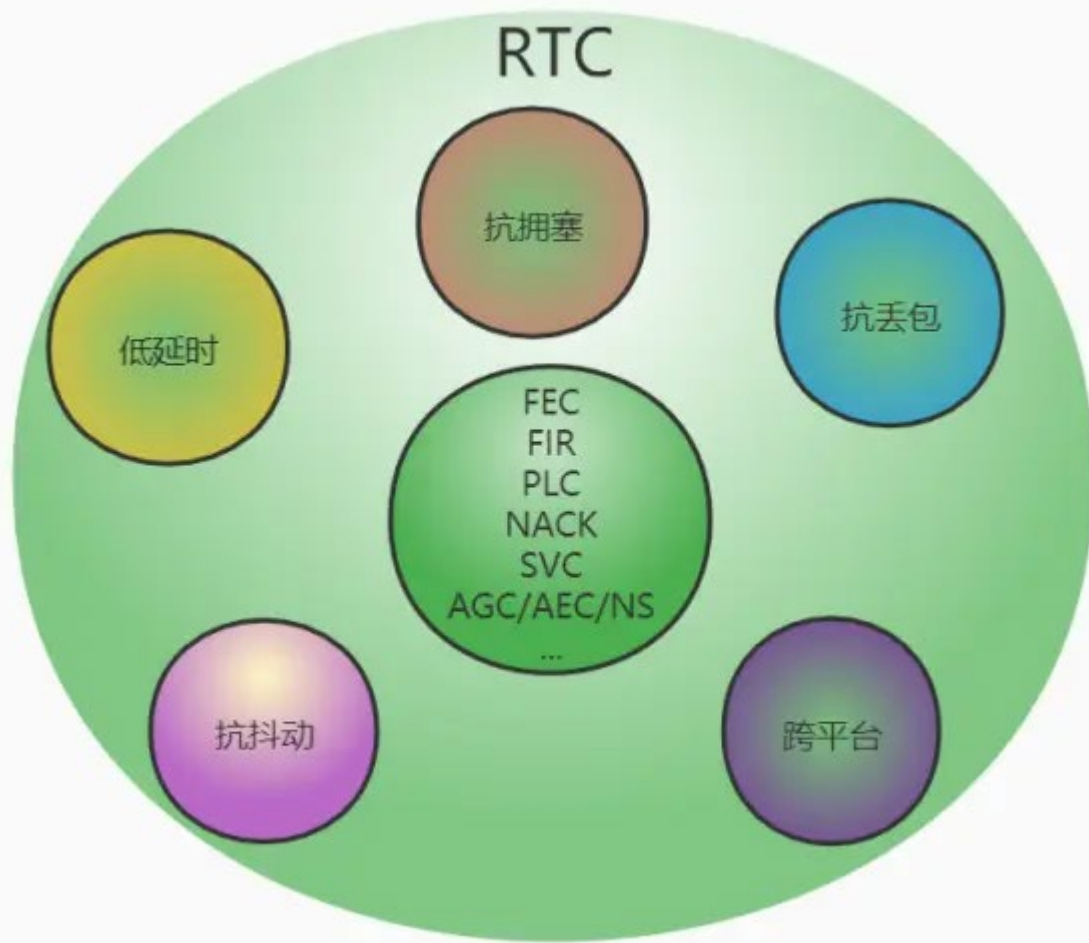


RTC直播系统架构



RTC技术要点

- FEC
- NACK
- JitterBuffer
- GCC
- Simulcast
- SVC

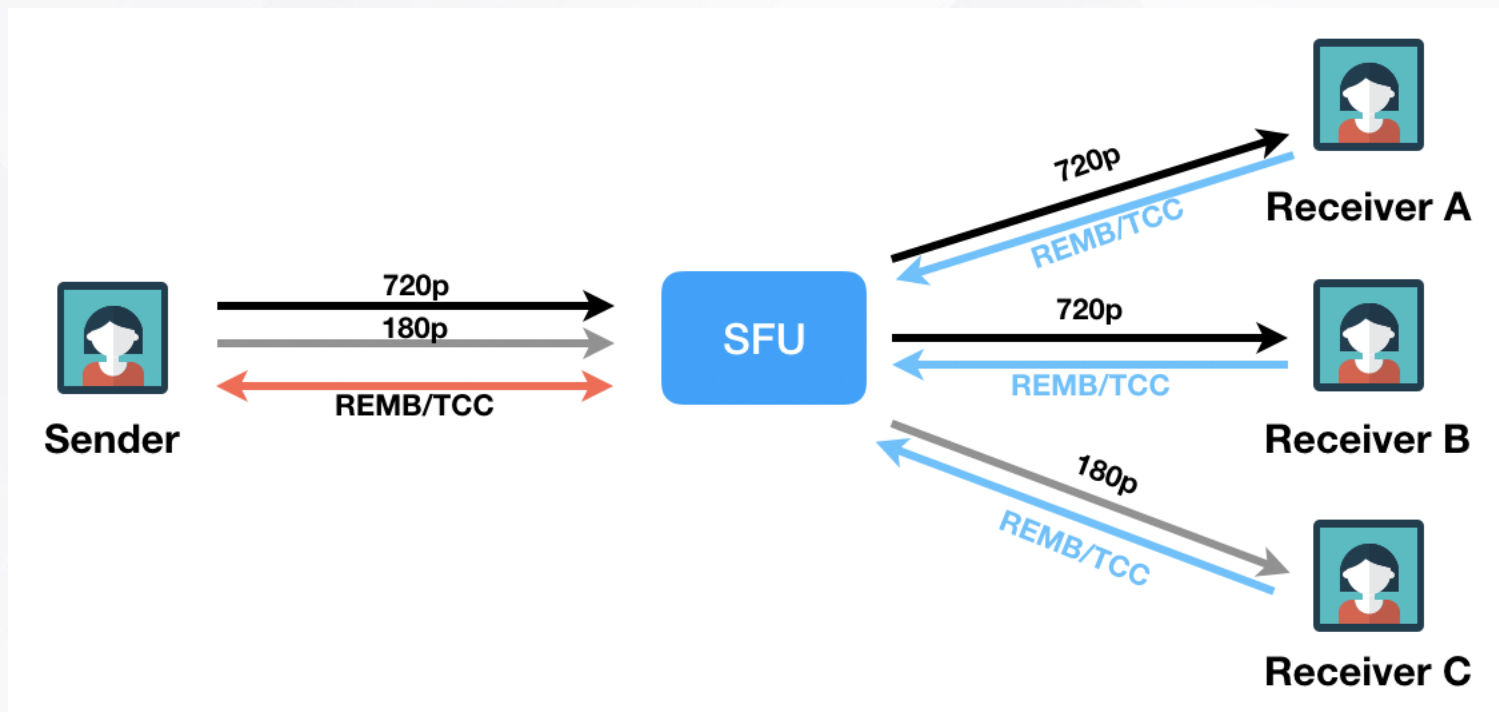


RTC技术要点

- Simulcast:

空间分层

时间分层



RTC技术要点

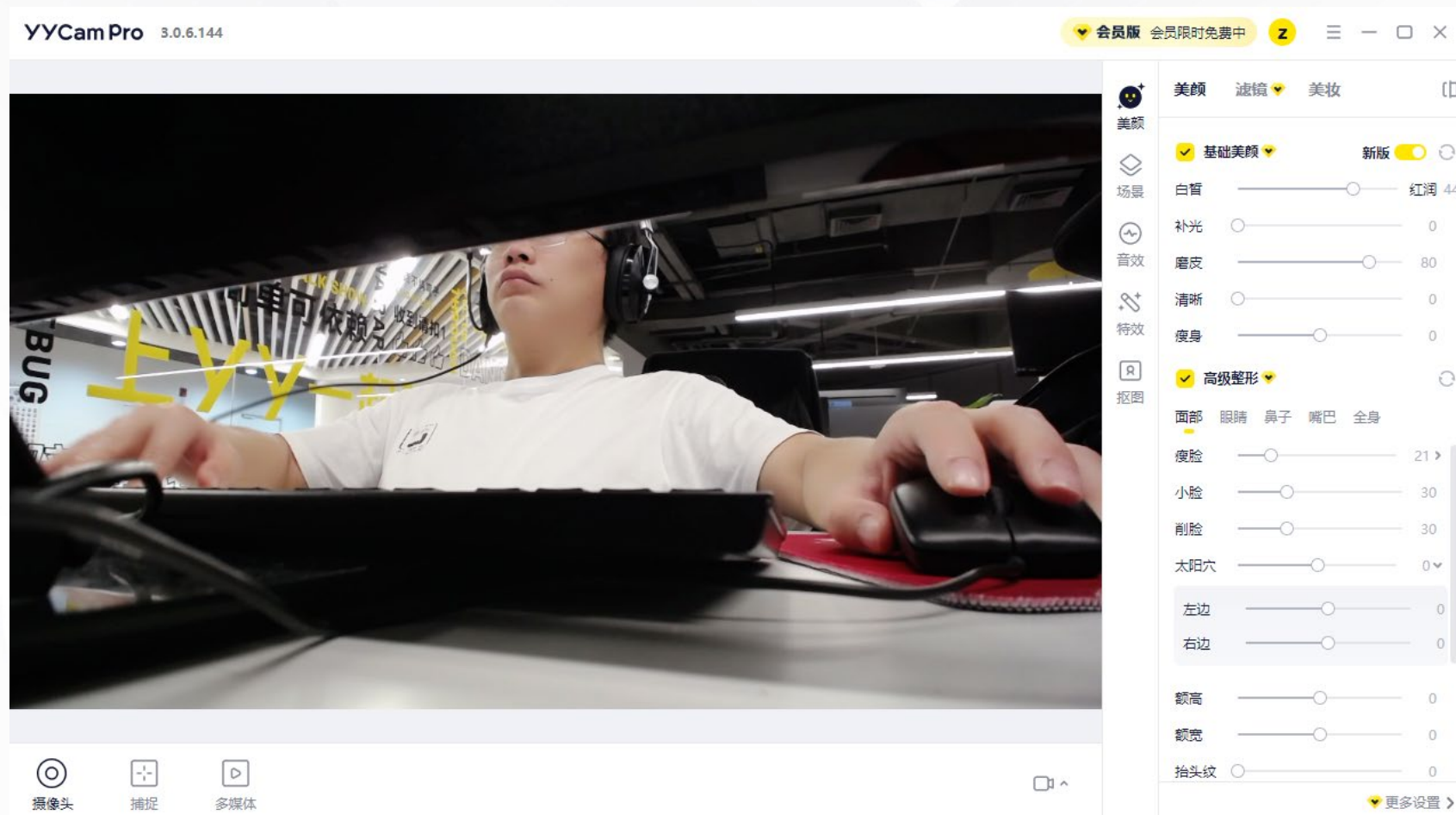
- **SVC:**

空间分层

时间分层

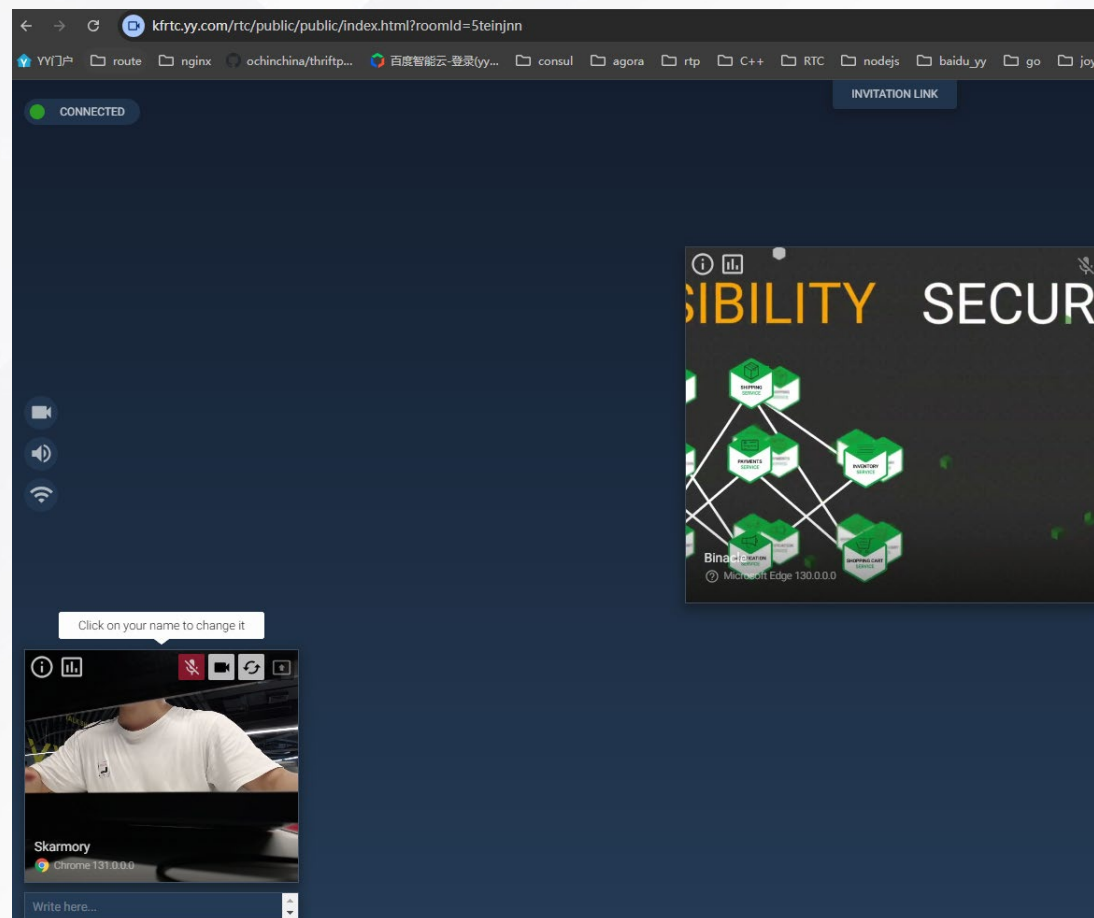
RTC-live-demo演示

利用开播工具采集音视频 -> webrtc推流 -> SFU -> MCU -> CDN



RTC-demo演示

2人视频通话



RTC-live-demo演示

直播流列表

🏠 控制台

💡 直播管理

在线直播

WebRTC混流

直播记录

在线主播

在线观众

在线stream

flv播放

转推列表

🏠 直播间管理

直播管理 / 在线stream

admin

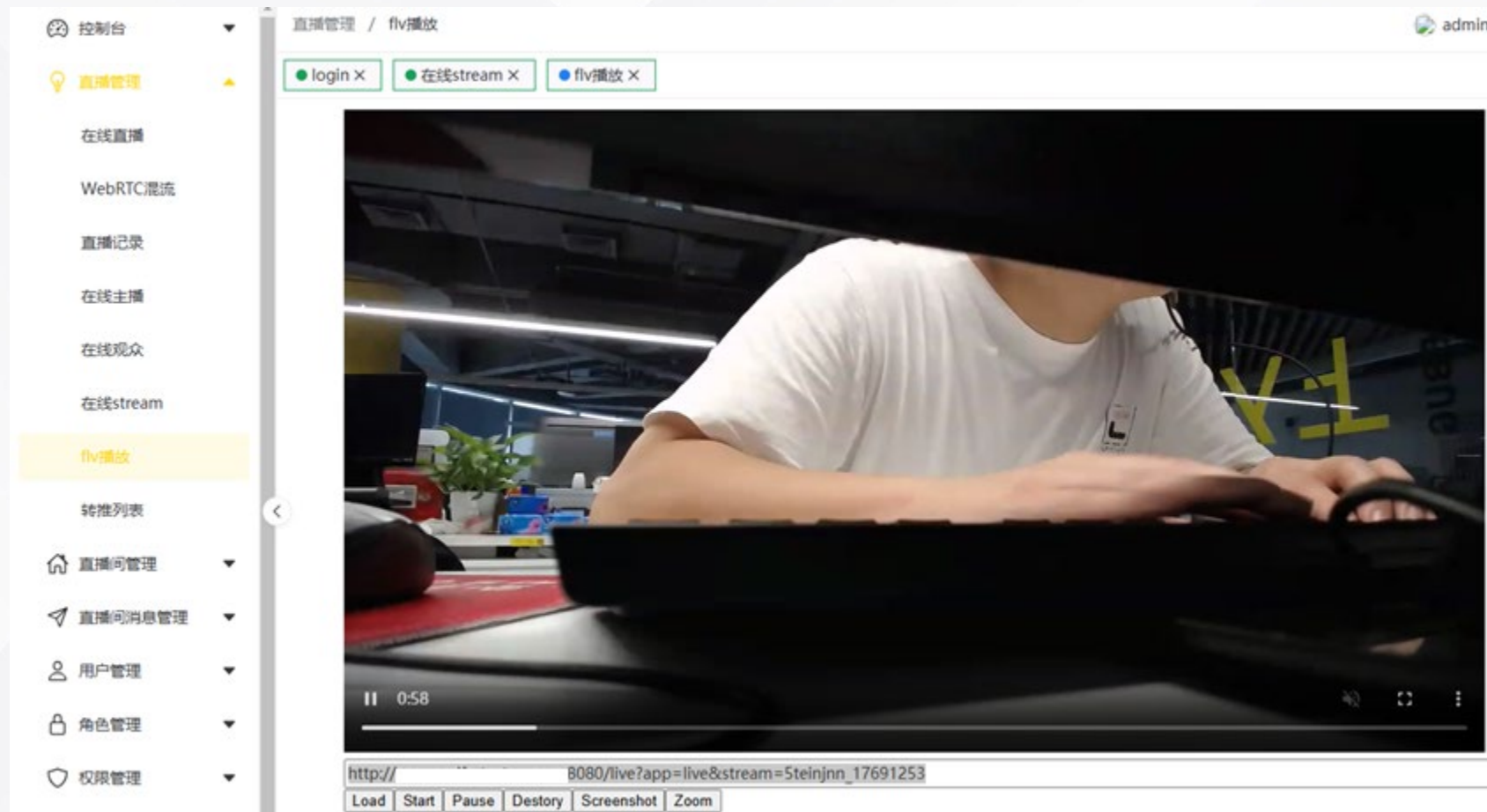
WebRTC配置 × token鉴权 × flv播放 × 在线stream × WebRTC混流 ×

分页大小: 20 当前页数: 1 当前页有几条记录: 3 上一页 下一页 刷新

| id | name | tcU | publish | clients | vhost | 操作 |
|----------|----------|---------|---------|---------|-------|----|
| 70726521 | xkxuvpoo | rtmp:// | http:// | | | 播放 |
| 25153523 | xkxuvpoo | rtmp:// | http:// | | | 播放 |
| 70726521 | xkxuvpoo | rtmp:// | http:// | | | 播放 |

RTC-live-demo演示

直播流http-flv播放



RTC-live-demo演示

Webrtc 转推rtmp直播流 转推配置

控制台

直播管理

直播间管理

直播间消息管理

用户管理

角色管理

权限管理

登录记录

全局消息

设置

直播配置

配置列表

转推配置

WebRTC配置

直播配置 / 转推配置

admin

转推配置 ×

room-yy2

uid

rtmp://

添加转推rtmpUrl

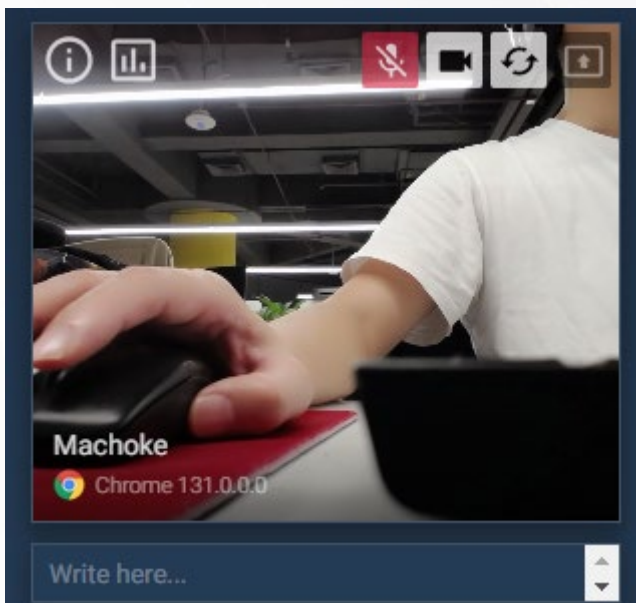
分页大小: 20 当前页数: 1 当前页有几条记录: 5 上一页 下一页 刷新

| roomId | uid | rtmpUrl | 操作 |
|---------------|-----|---------|-------|
| room-huya | | rtmp:// | 更新 删除 |
| room-yy | | rtmp:// | 更新 删除 |
| room-yy2 | | rtmp:// | 更新 删除 |
| room-bilibili | | rtmp:// | 更新 删除 |
| room-douyin | | rtmp:// | 更新 删除 |

1. 转推到第三方平台: YY/抖音/虎牙/b站
2. 转推到自建cdn, 如nginx-rtmp/srs
3. 转推到CDN

RTC-live-demo演示

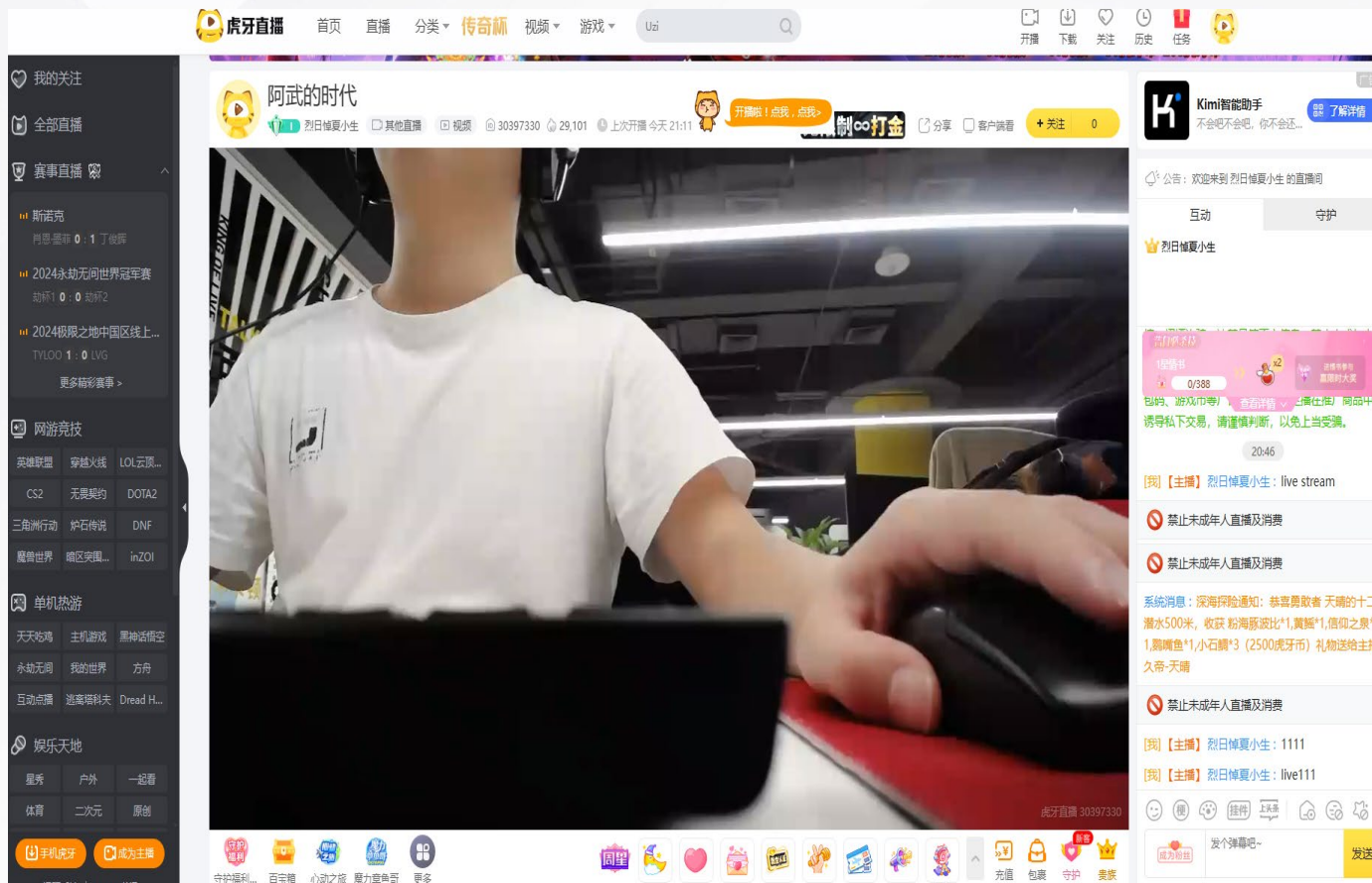
Webrtc 转推rtmp直播流到虎牙直播间



rtc -> rtmp

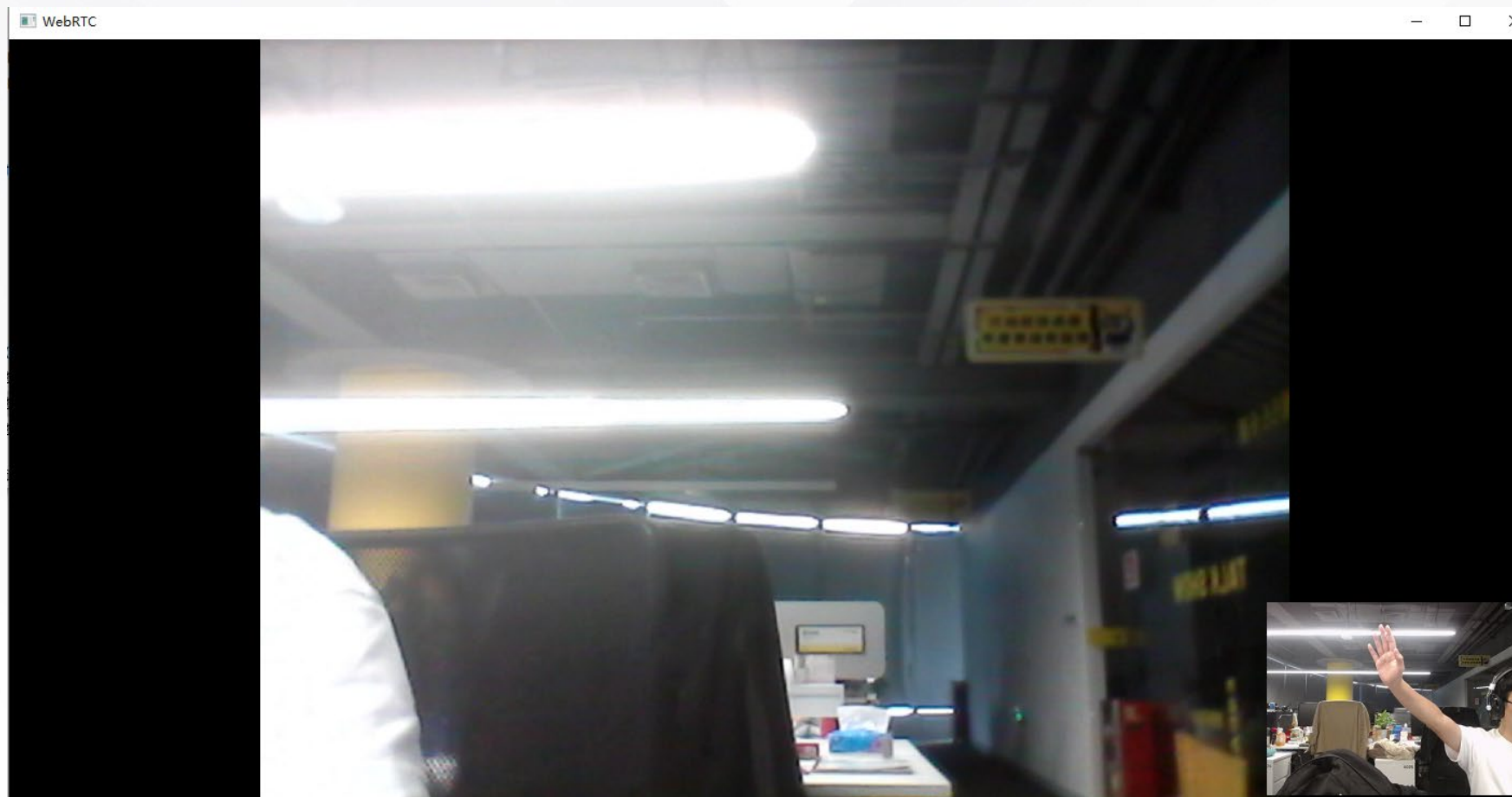


推流端：Webrtc推流



观看端：http-flv或hls流

WebRTC-P2P通话demo



RTC-mixer-RTMP

2路rtp实时流做混画（画中画）



RTC-mixer-RTMP

Webrtc -> sfu -> mcu -> rtmp 2路rtp实时混流（画中画）



优势1

1. 技术能力

- * P2P通话
- * 单人直播 -> 多人连麦直播
- * 接入CDN（灵活接入多家厂商的CDN，并进行流量调度）

2. 优势

- * 低成本； -- 在夹缝中求生存，生成中找机会
- * 系统稳定

优势2

优势2

1. 轻资产：直接给需求方 部署在私有 机器 或 云机器

- * 轻资产，技术优先：

- * 不依托 其他第三方的媒体处理云服务（转码，录制）。-- 只使用对象存储，CDN网络分发。

2. 成本低：

夹缝里求生存， 小的电商需要使用客服；

小的公司需要使用客服；

小的教育公司需要在线直播；

-- 小的创业公司不一定能够得到 声网 的优质VIP服务。 但是我们能够提供稳定的服务，并及时解决问题。

3. 防攻击；

- * 接入时账号安全；

- * 系统加固，安全稳定

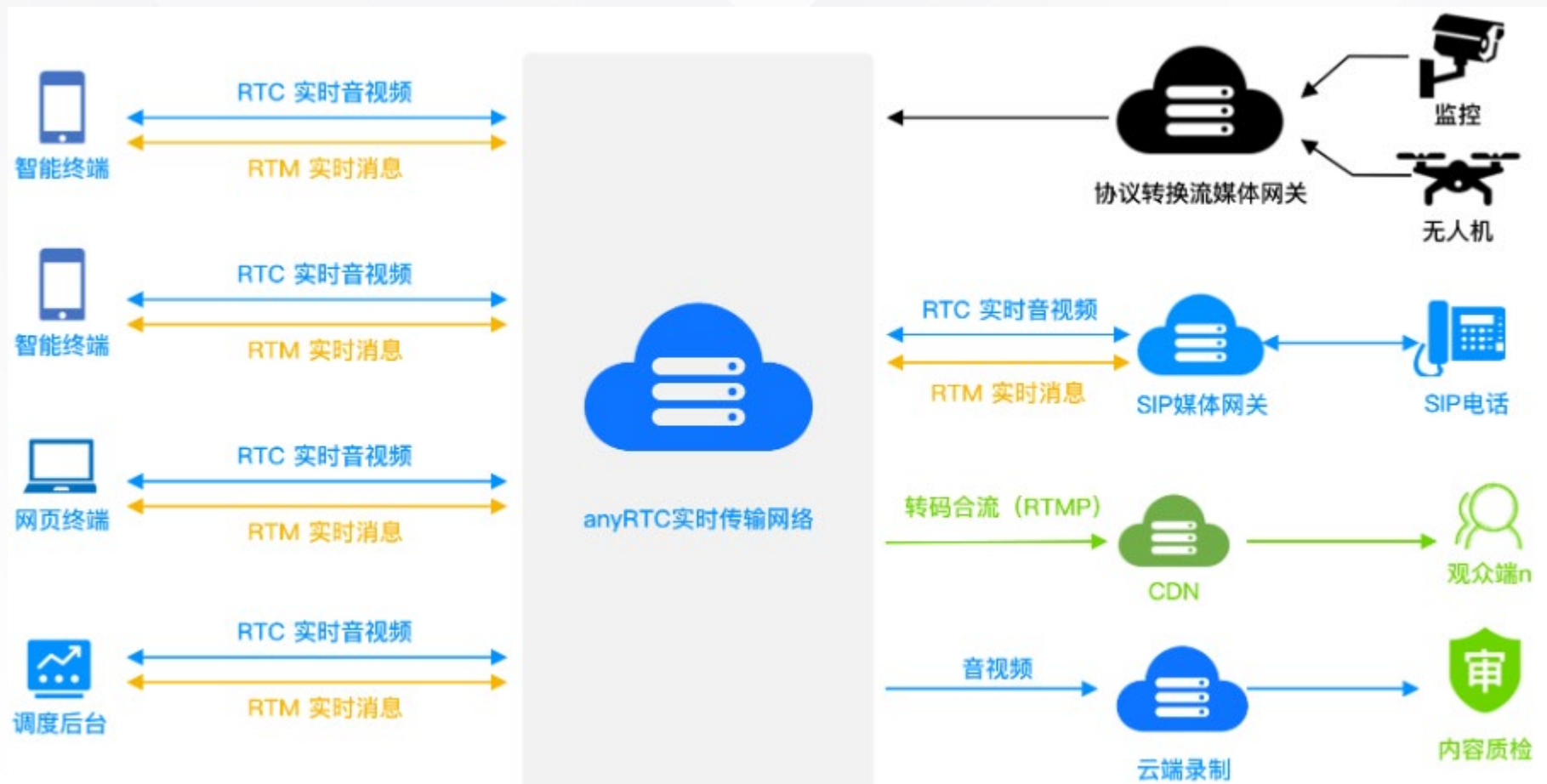
WebRTC一站式直播解决方案

FAQ

1. 没什么不直接使用ffmpeg命令做RTC直播系统的mixer混流？

- * 性能没有ffmpeg c++二次开发的效果好；c++二次开发 专门针对rtp->rtmp做了性能优化，性能更优：出流快，延迟低；
- * c++程序安全性（部署、优化参数/input/output等不会暴露）更好，更利于打造自己的核心竞争力；
- * 第1阶段：ffmpeg支持的指令不能直接满足接受多条RTP流，需要二次开发；
- * 第2阶段：ffmpeg拉流效果不好，丢包等需要深度修改ffmpeg rtp/rtcp代码；
- * 第3阶段：自研rtp/rtcp拉流的逻辑，让ffmpeg c++做编解码相关；-- 达到预期效果；

音视频解决方案

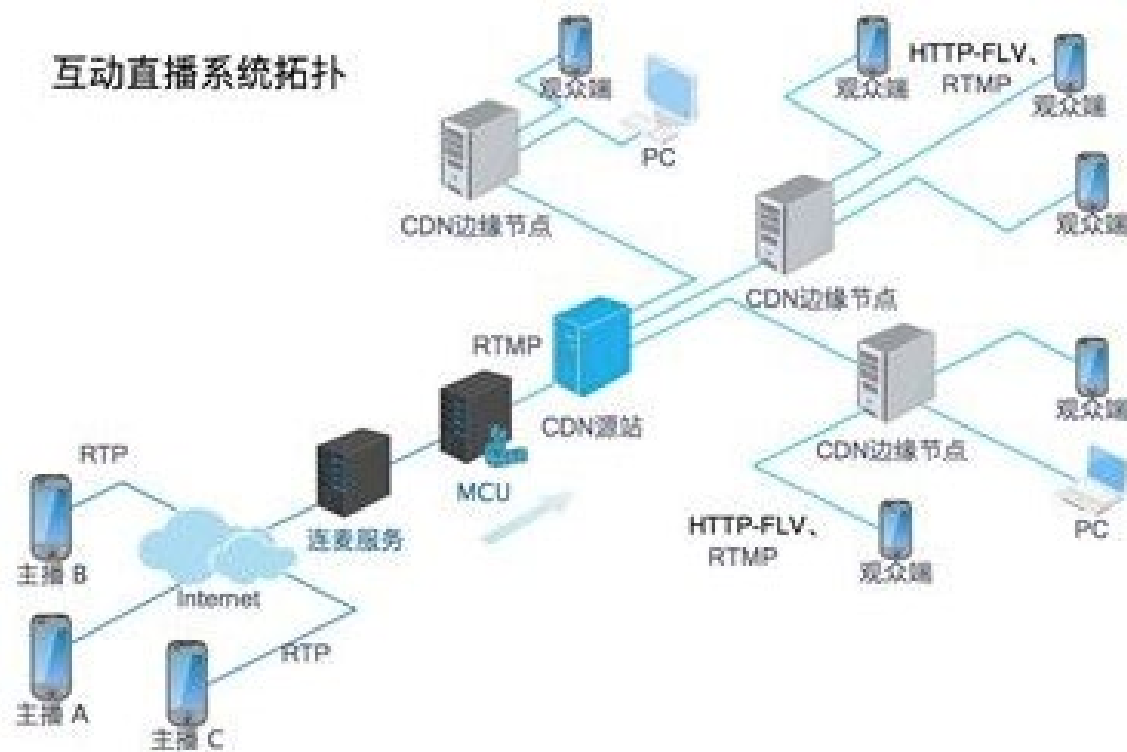


音视频解决方案

CDN直播系统拓扑



互动直播系统拓扑



FAQ

个人小结:

- 喜欢coding, 工作和业余写了很多代码, 并且尽量与优秀开源对齐
- 根据情况会一些技术, 代码开源出去

Github:

<https://github.com/cherishman2005/nginx-modules>

<https://github.com/cherishman2005/rtc>