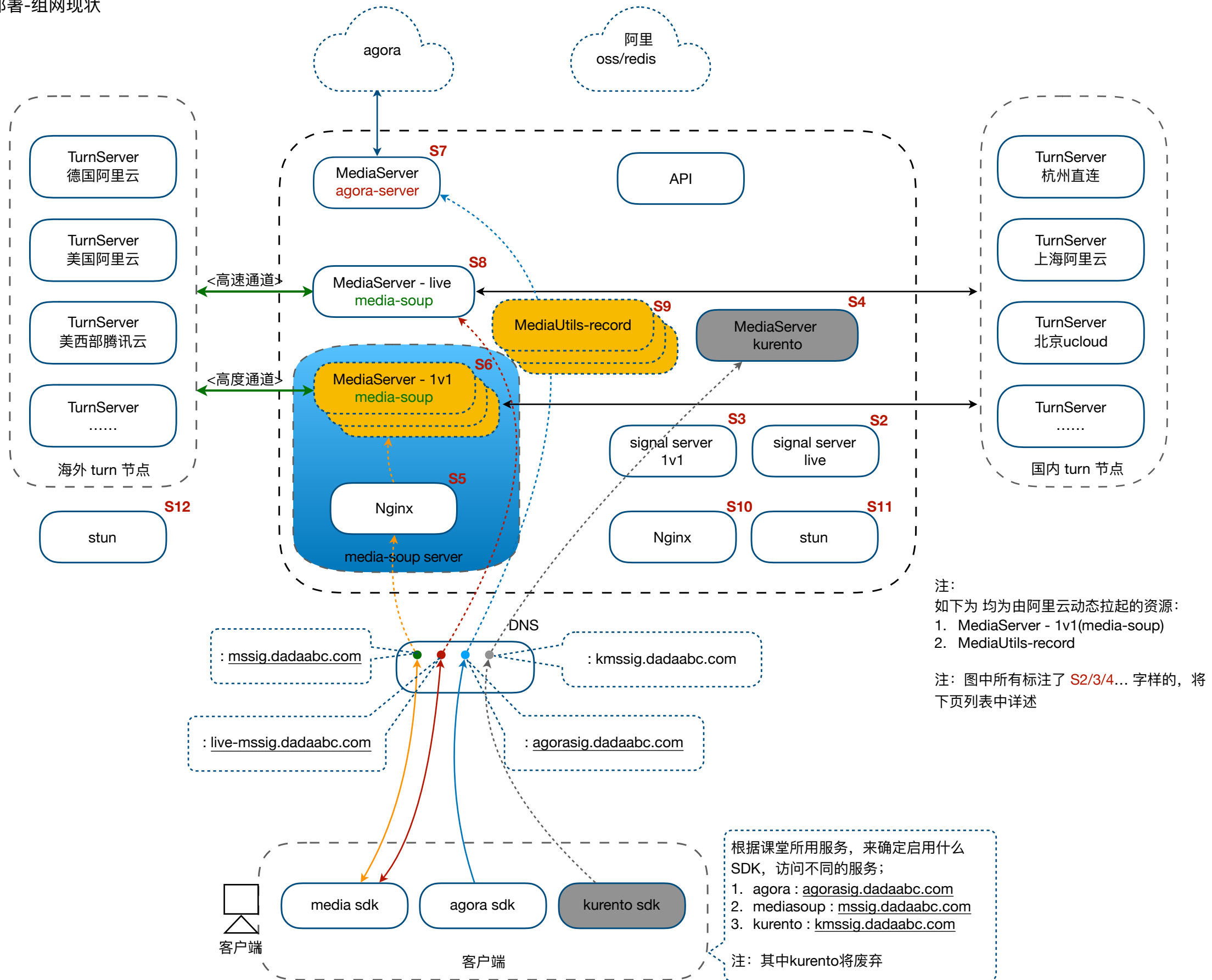


# 演进的思路

- 线上组网部署 — 现状
- 组网的演进
- 视频能力的演进
- 落地任务
- 未来思考

# 线上部署组网 — 现状

- 线上部署-组网现状



- 线上部署-各服务域名 — 从 `classroom/env/production.php` 中获得

标号		说明	域名
S1	SIG_SERVER SIG_SERVER1	直播课 <a href="#">socket.io</a> — 除音视频之外的其他信令/数据类通信	<a href="#">sigserver.dadaabc.com:6060</a> <a href="#">sigserver1.dadaabc.com:6060</a>
S2	LIVE_MQTT_HOST LIVE_MQTT_HOST1	直播课 mqtt — 仅仅用于信息转发，无任何逻辑 注：用于直播课内的 画板操作、文字聊天、通知等数据类通信	<a href="#">livesig.dadaabc.com:8443</a> <a href="#">livesig1.dadaabc.com:8443</a>
S3	V3_MSG_SIGNAL_HOST V3_MSG_SIGNAL_HOST1	1v1课 mqtt — 除音视频之外的其他信令/数据类通信（如画板操作、文字聊天、通知等）	<a href="#">classmq.dadaabc.com:8088</a> <a href="#">classmq1.dadaabc.com:8088</a>
S4	V3_KURENTO_SIGNAL_HOST V3_KURENTO_SIGNAL_HOST1	1v1课 kurento 音视频信令/媒体服务器	<a href="#">kmssig.dadaabc.com:8086</a> <a href="#">kmssig1.dadaabc.com:8086</a>
S5	V3_MEDIASOUP_SIGNAL_HOST V3_MEDIASOUP_SIGNAL_HOST1	1v1课 mediasoup 音视频信令服务器入口（nginx）	<a href="#">mssig.dadaabc.com:8087</a> <a href="#">mssig1.dadaabc.com:8087</a> 注：指向 <i>nginx</i> — 其后为 媒体服务器
S6		mediasoup 音视频媒体服务器	不采用域名访问 注：阿里云动态拉起的云主机
S7	V3_AGORA_SIGNAL_HOST V3_AGORA_SIGNAL_HOST1 V3_LIVE_AGORA_SIGNAL_HOST V3_LIVE_AGORA_SIGNAL_HOST1	agora 服务器 注：仅用于生成key	<a href="#">agorasig.dadaabc.com:8002</a> <a href="#">agorasig1.dadaabc.com:8002</a>
S8	V3_LIVE_MEDIASOUP_SIGNAL_HOST V3_LIVE_MEDIASOUP_SIGNAL_HOST1	直播课 mediasoup 音视频信令/媒体服务器	<a href="#">live-mssig.dadaabc.com:9001</a> <a href="#">live-mssig1.dadaabc.com:9001</a>
S9		音视频录制服务器	不采用域名访问 注：阿里云动态拉起的云主机
S10	杭州直连	杭州直连，客户端测速访问点 注：在音视频部分，专作为客户端到杭州直连节点测试流量；	121.196.220.124
S11	stun	客户端网络测速后，若选中为 杭州直连 时，ice 采用 stun	121.40.61.36 注：国内 — 阿里云杭州机房
S12	stun	客户端网络测速后，若选中为 杭州直连 时，ice 采用 stun	47.88.10.123 注：海外 — 美国阿里云
	REDIS_HOST	redis	<a href="#">r-bp1cf98b3ff50b04.redis.rds.aliyuncs.com</a>
	LIVE_NODE_API LIVE_NODE_API1	直播课的在线列表获取接口 注：目前仅仅用于此功能	<a href="#">https://livesig.dadaabc.com:443</a> <a href="#">https://livesig1.dadaabc.com:443</a>

• 各边缘节点&流量分布情况 — 海外 — 数据待补充

标号	名称	IP	流量	CPU\MEM
TA1	德国阿里云			
TA2	德国阿里云1			
TA3	德国阿里云2			
TA4	德国阿里云3			
TA5	德国腾讯云			
TA6	美国阿里云			
TA7	美国阿里云1			
TA8	美国西部腾讯云			
TA9	美国东部腾讯云			
TA10	日本阿里云			
TA11	香港阿里云			
TA12	香港阿里云1			
	香港腾讯云			

• 各边缘节点&流量分布情况 — 国内 — 数据待补充

标号	名称	IP	流量	CPU\MEM
TB1	北京阿里云			
TB2	北京ucloud			
TB3	上海阿里云			
TB4	上海市阿里云2			
TB5	上海市阿里云3			
TB6	上海市阿里云4			
TB7	上海市阿里云5			
TB8	深圳阿里云			
TB9	腾讯广州			
TB10	腾讯成都			
TB11	青岛阿里云			
TB12	张家口阿里云			
TB13	湖南电信			
TB14	新数讯			

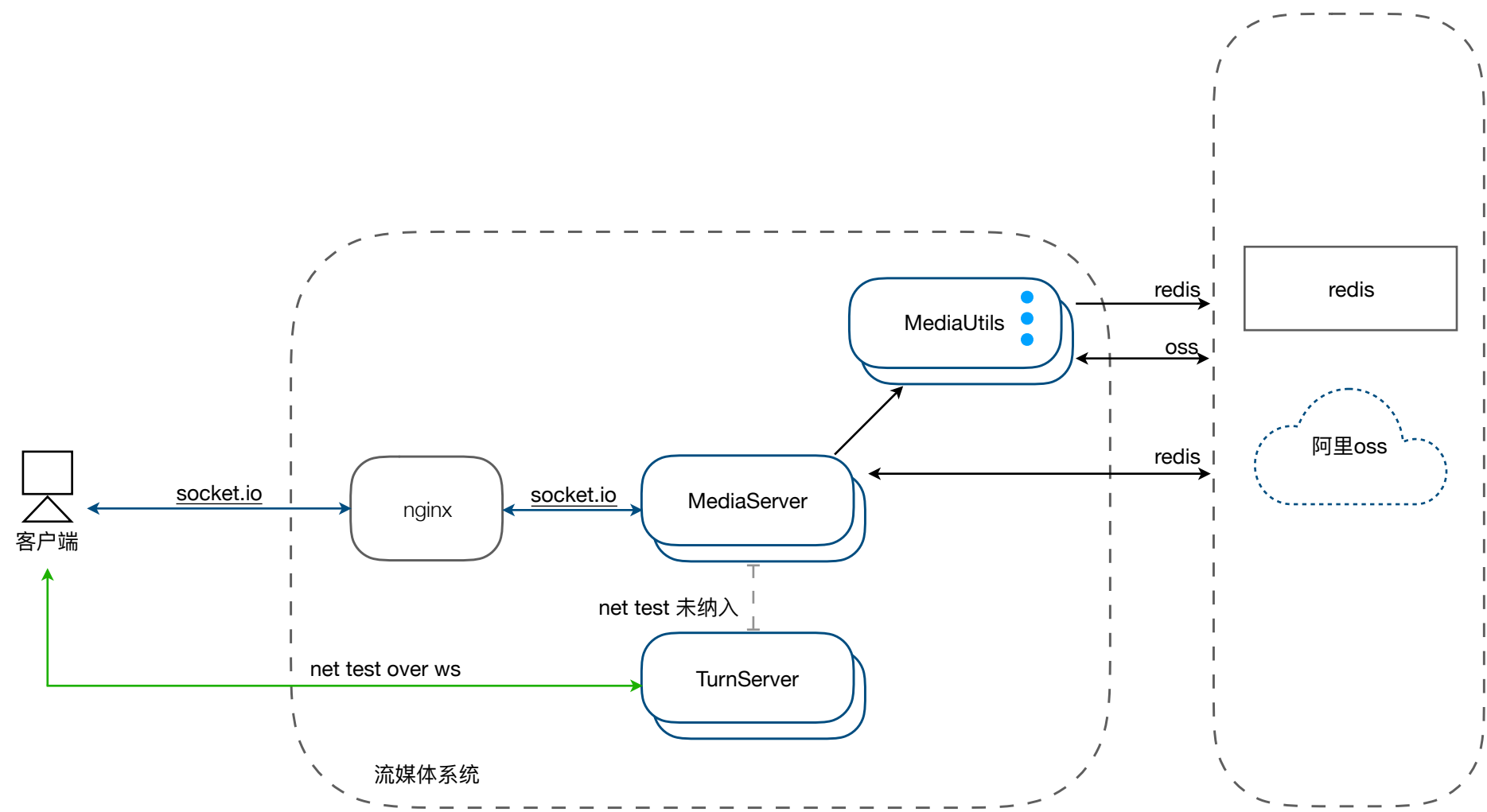
- 分析

待定

# 组网演进



• 当前架构



MediaServer	sfu, 基于 MediaSoup 实现
TurnServer	基于 coturn 实现
MediaUtils	录制工具集, 基于 nodejs + ffmpeg 实现

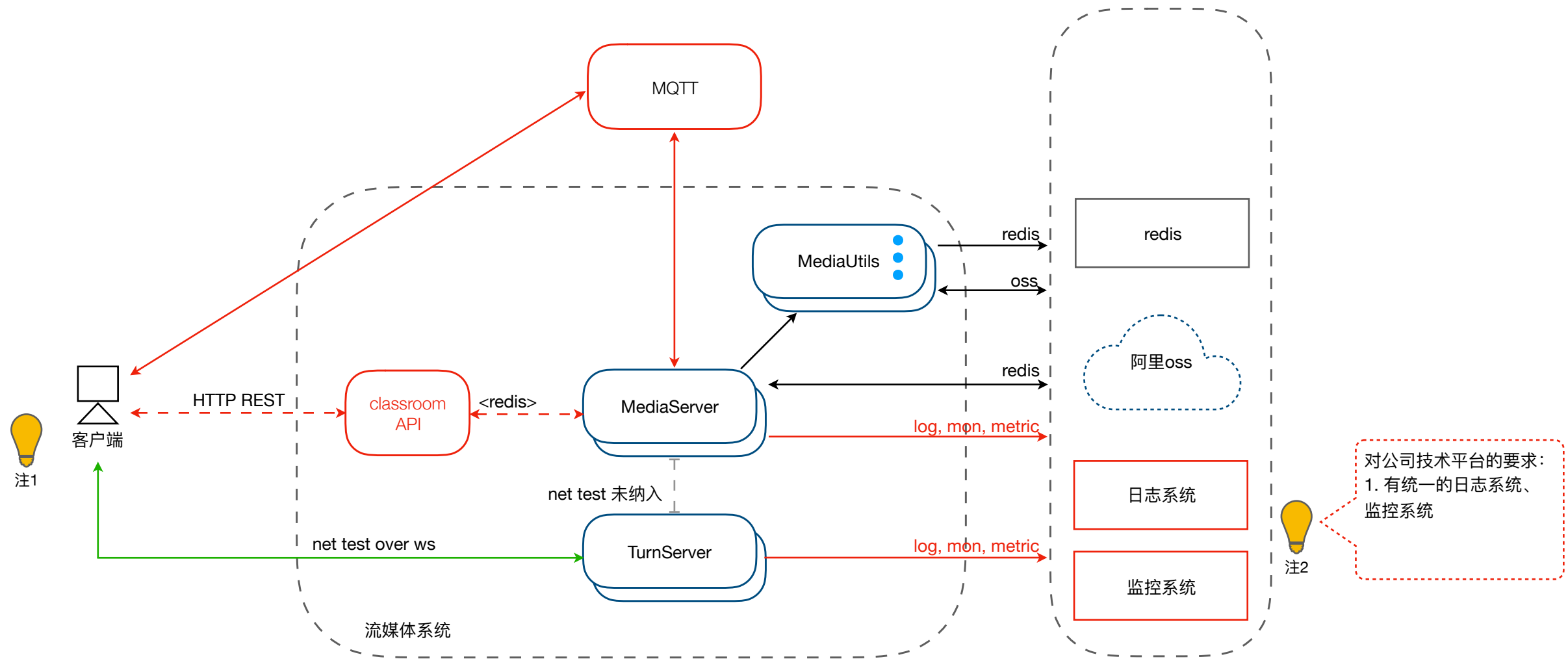
- 演进1
  1. 提供classroom API, 获取适合的MediaServer
  2. 客户端到系统, 采用MQTT交互信令
  3. MediaServer、TurnServer均提供媒体传输状态统计及度量

注1: 多种类型客户端均涉及

注2: 对公司技术平台的要求 — 有统一的日志系统、监控系统

解决的问题:

1. 无metric, 无更细节的业务监控;
2. 未依据 MediaServer 的负载来分配
3. 未监控 MediaServer、TurnServer 的负载情况
4. 没有 MediaServer、TurnServer 的相关运行日志备查



- 度量指标

分类		详
指标	流媒体传输状态	活动链接数（通道）
		每通道的音视频编码、丢包率、抖动、延迟；
		每通道的视频分辨率、码率；
		建链（ICE）失败率、中断率；
	系统资源	总带宽（出/入）
		CPU/MEM（特定进程占用，如turn server、media server）
	网络监测（含测速）	客户端网络测速：RTT、丢包率、抖动、带宽（尝试冲击带宽）
		边缘节点 <—> 核心节点：RTT、丢包率、抖动
来源（对象）		Turn server、Media server、客户端；

注1：服务器以IP为ID标识；客户端以学生ID为标识；  
注2：度量信息的记录：时间、来源（对象）、来源（对象）ID、指标；  
注3：监控 — 定时上传（定时间隔可定义，默认为 10s）；数据可视化；

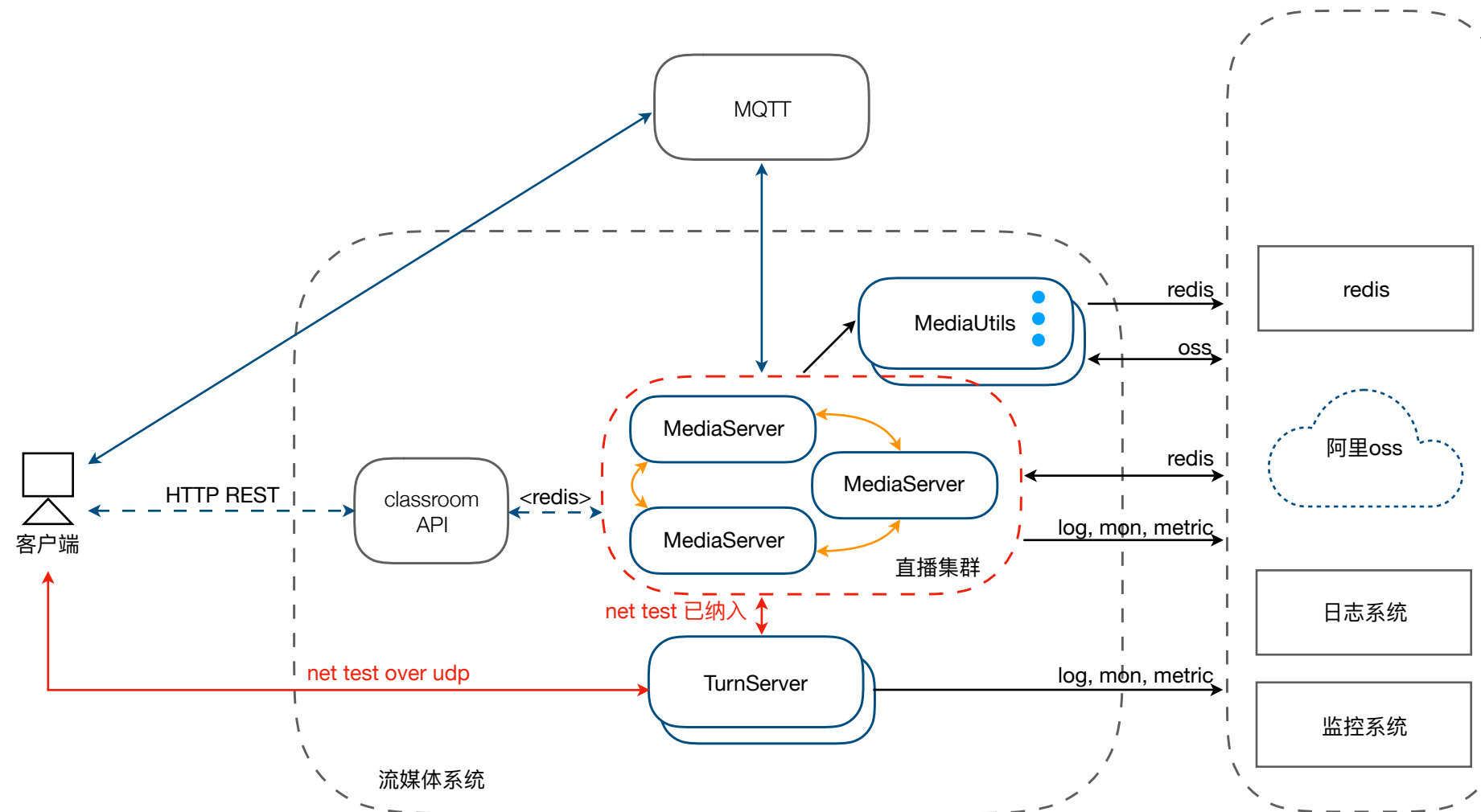
- 演进2

1. MediaServer的直播集群
2. 客户端到系统的网络测速

注：网络测速 — 采用 over udp 的方式，并将  
turnserver - mediaserver段也纳入；丰富探测指标

解决的问题：

1. 直播课 的容量有限
2. 网络测速路径覆盖不全



注1：【度量指标】中有针对 客户端网络测速的指标定义

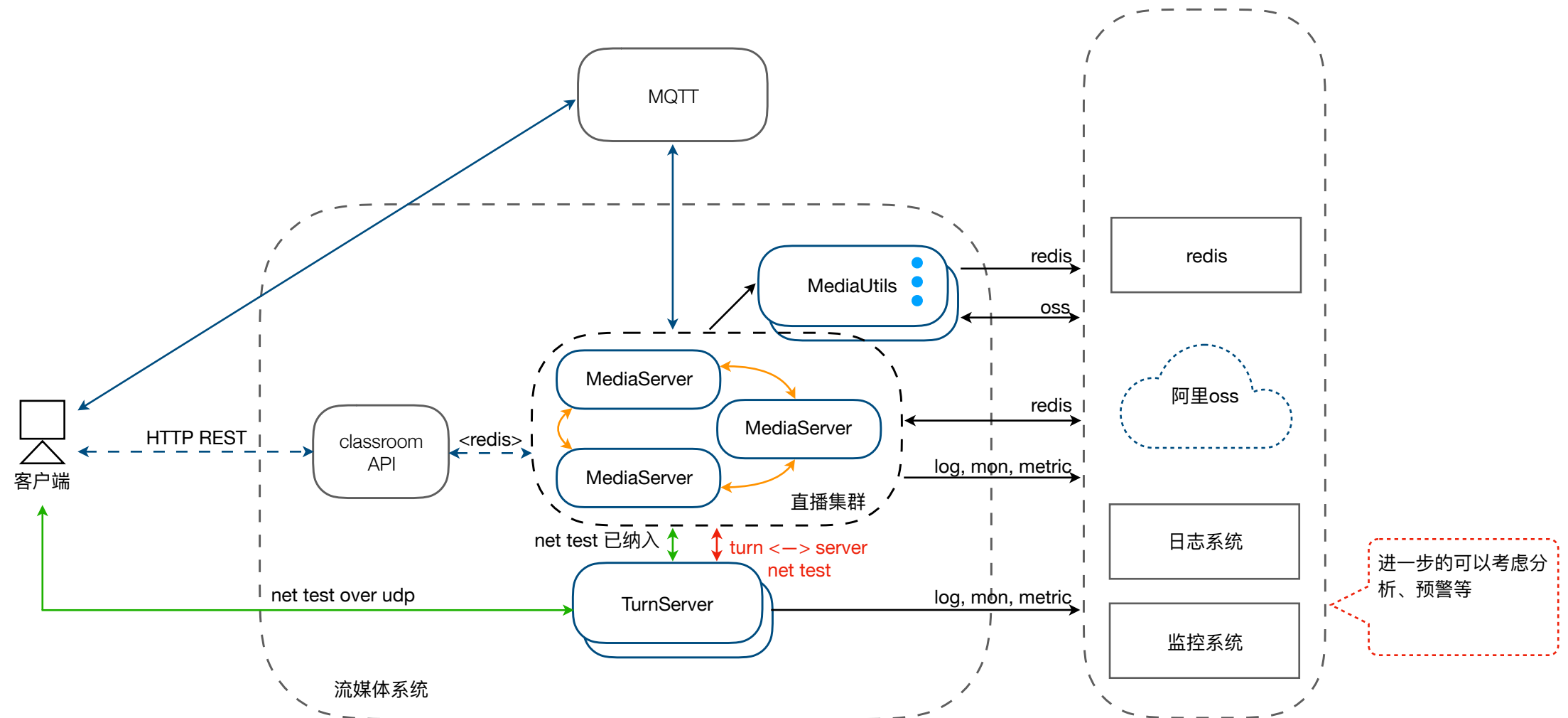
- 演进3

1. TurnServer - MediaServer 间的网络探测

注：网络测速 — 采用 over udp 的方式，并将  
turnserver - mediaserver段也纳入；丰富探测指标

解决的问题：

1. 对 TurnServer 到 MediaServer 之间链路异常的时无预警



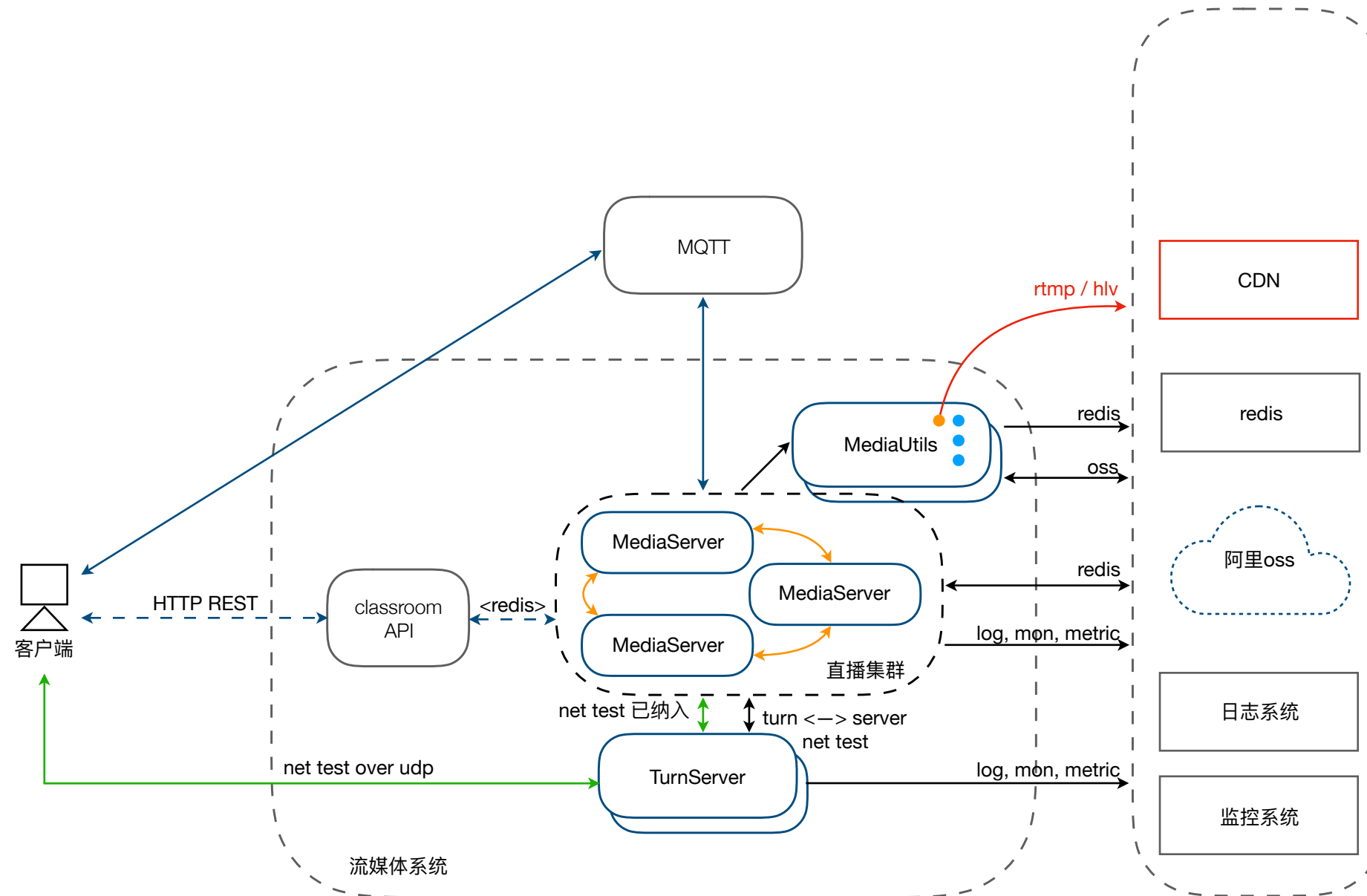
- 演进4

- 1. 推流 live stream -> CDN

注：通过CDN的方式，进一步扩展直播课的容量

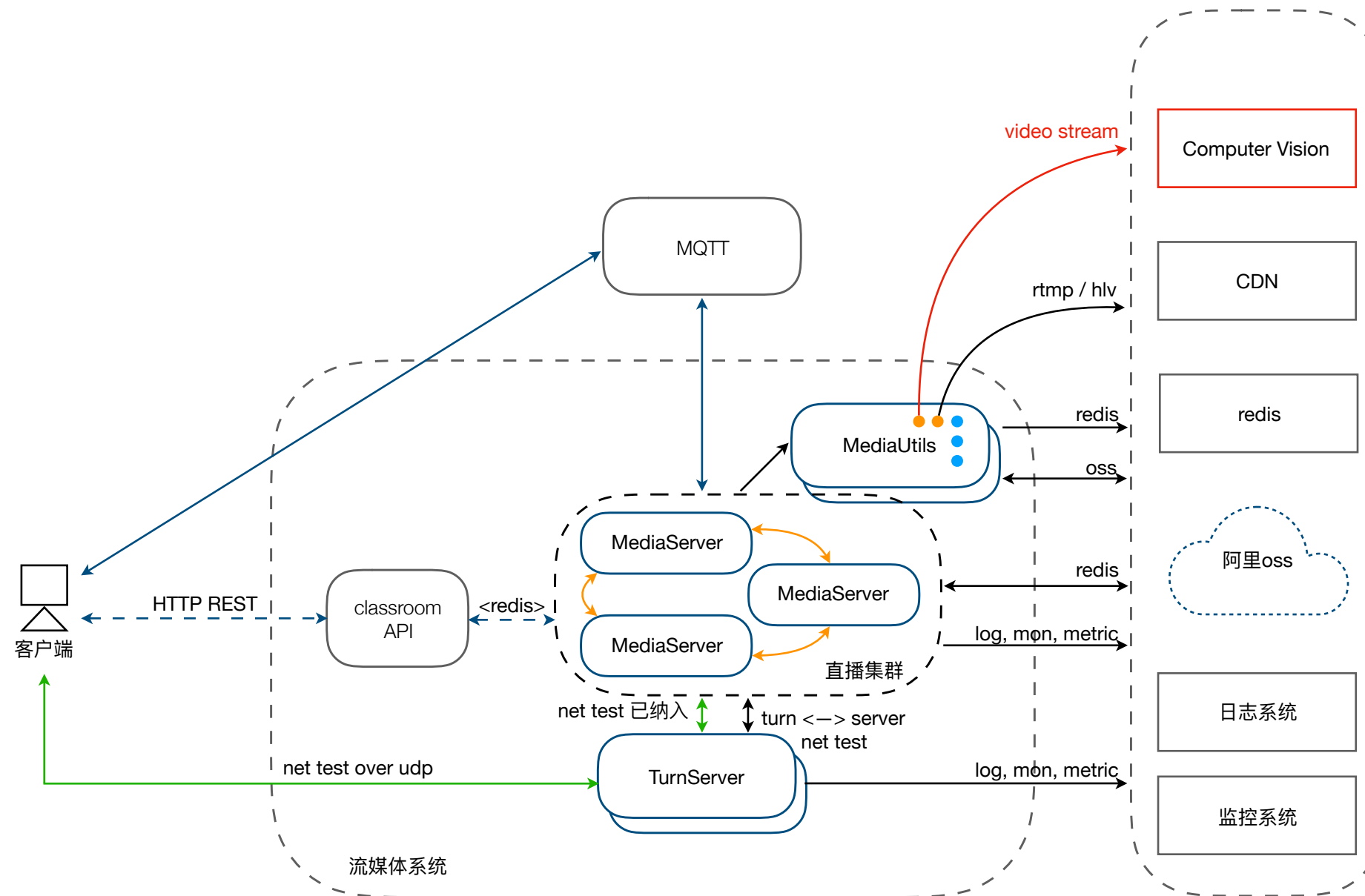
解决的问题：

- 1. 直播课 大容量用户时，流量成本过高



- 演进5 — 未来

- 1. 视频分析/处理 — 可能场景：图像识别、合规鉴定、.....



- 演进6 — 未来

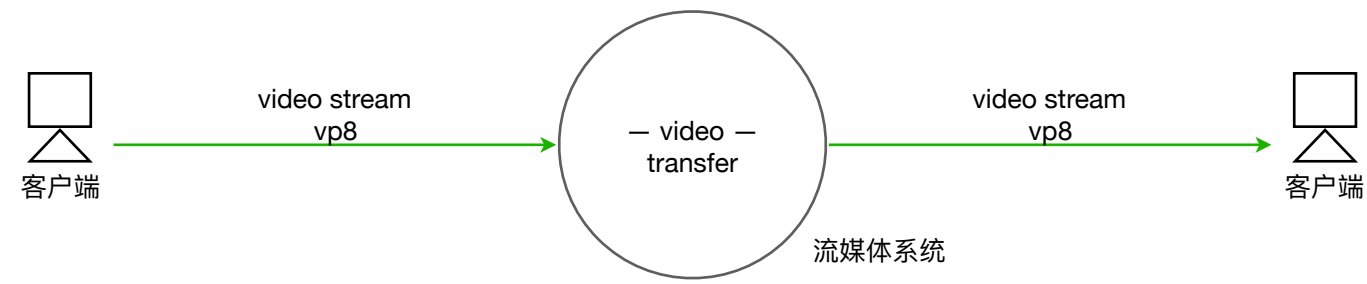
Media Server SDN?  
依据应用场景待定



视频能力

- 现状

1. 视频采用vp8编码；MediaServer 对视频流采用直接中转的方式



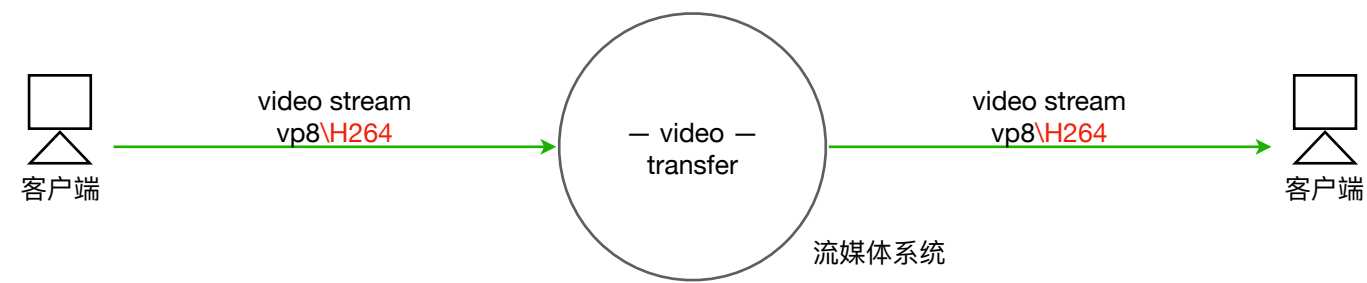
- 演进1

1. 视频编码支持H264

注：考虑在各端上H264 HighProfile 的支持情况的测试验证；

解决的问题：

1. 终端编解码的CPU耗用过高，功耗高



- 演进2

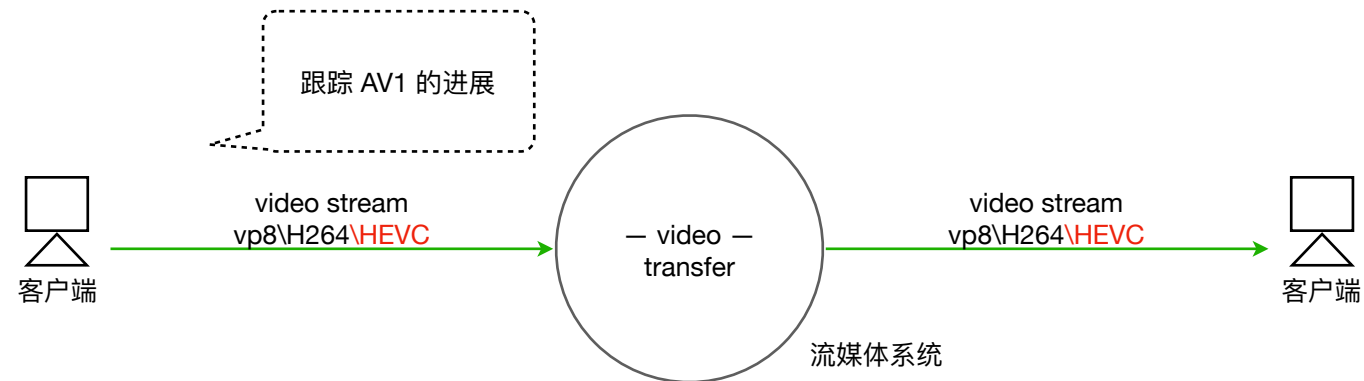
1. 视频编码支持HEVC

注：考虑在各端上HEVC 的支持情况的测试验证；

注：目前在iOS 11 系统 VideoToolbox已支持HEVC，在iPhone 7以上平台，已支持HEVC硬编解码；

解决的问题：

1. 合适码率的情况下，清晰度不够



- 演进3

1. 1对多场景下，mediaserver应按不同端能力提供 【差分服务】
2. 来自多端的 RTCP feedback，经 MediaServer 汇聚处理，综合之后再反馈源端

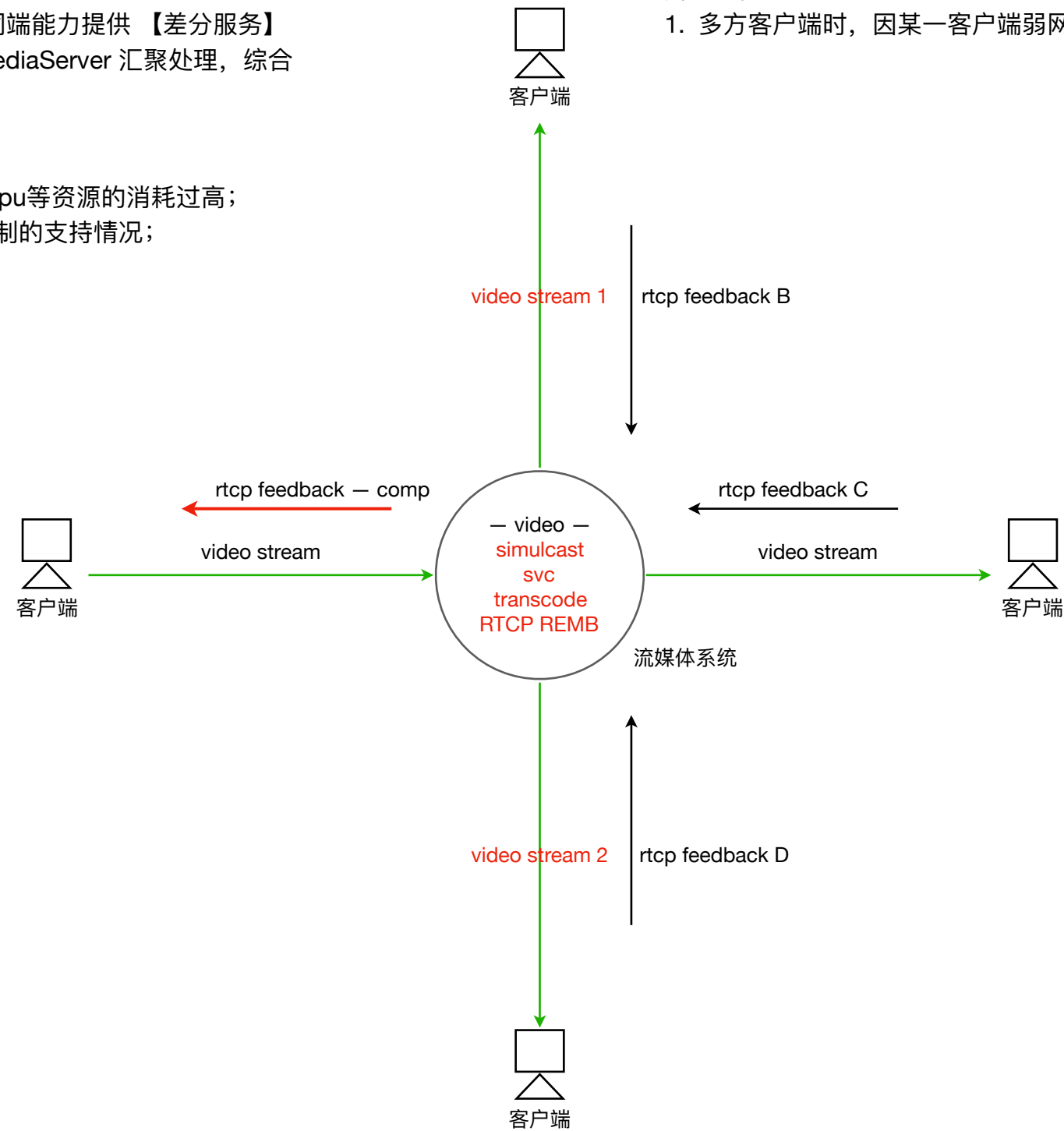
注：如 simulcast、svc 等方案；

注：transcode也是一种方案，但其对cpu等资源的消耗过高；

注：具体验证一下openh264的svc机制的支持情况；

解决的问题：

1. 多方客户端时，因某一客户端弱网而导致所有客户端视频质量差



# 落地任务

- 开发

序号	任务	状态	备注
1	录制流程优化	进行中	2018.4.23: 目前已提测
2	直播课实现	进行中	2018.4.24: 开发进行中
3	度量、监控		媒体状态统计/记录（包括传输、活跃对象等） 包括 MediaServer、TurnServer、客户端；
4	网络测速 / server 选择		
5	TurnServer <—> MediaServer 网络状况探测		
6	SFU 下多端时RTCP汇聚处理（拥塞控制相关）		
7	录制流程优化 - 迭代优化2		
8	具体优化点：视频带宽范围（最大、最小带宽）		
9	具体优化点：视频FEC支持		
10	音视频媒体能力的环回测试		
11	线上kurento废弃		注：应需要客户端上层逻辑来处理
12			

- 预研/技术准备

序号	任务	状态	备注
1	webrtc 中 openh264 svc 研究		
2	webrtc 中 simulcast 验证		
3	webrtc 断线重连机制预研（PeerConnection层面）		
4	HEVC支持		
5	AV1跟踪		
6	流量监测与Qos评估		



未来思考

1	AR 在实时课堂中的应用
2	全景视频（360）、3D在实时课堂中的应用