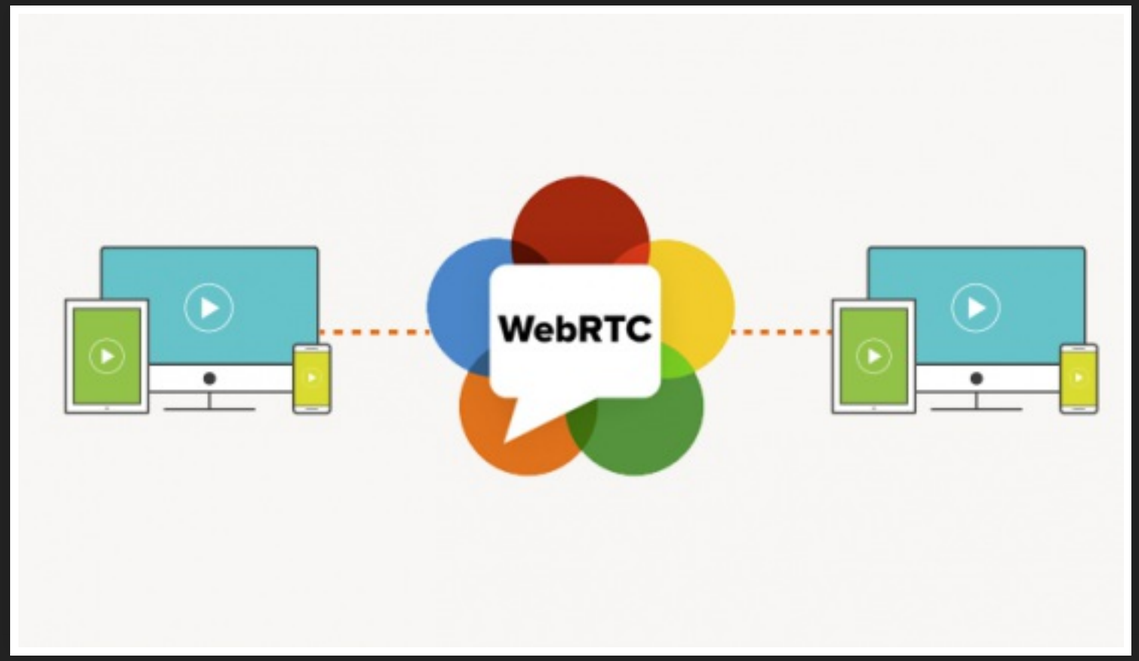


基于 WEBRTC 技术的实时通信服务开发实践

刘博 @ UPYUN

2017-07-08

OUTLINE

- 什么是互动直播?
- 为什么选择 WebRTC.
- WebRTC 的技术特点.
- 底层技术平台.
- 我们的一个应用场景 - 上会.
- 我们的一些开发经验总结.

互动直播是多路音视频以及数据实时通信的解决方案.

首先, 看一下我们又拍云自己的应用场景.

- 又拍云是由杭州研发总部加上北京, 上海, 广州等多家分公司组成的.
- 我们的管理层每周一上午有一个每周例会. 我们很需要一个简单可靠的远程会议系统.

在过去, 我们用的是传统的电话会议硬件. 复杂难用, 价格也不便宜, 而且效果也不理想.

现在, 我们只需要一台笔记本电脑外加一个高清摄像头就可以了. 不需要任何配置, 安装任何插件.

互动直播的特点

- 互动直播和传统直播相比的本质的区别是延时.
- 与传统的直播相比, 互动直播赋予了参与直播的每一个成员互动交流的能力, 因此, 也对实时性, 抗回声要求更高.
- 在视频会议, 远程教育, 远程咨询, 视频社交, 互动游戏等很多场景往往只能选择实时性更高的互动直播技术.

为什么选择 WEBRTC?

- WebRTC 是一个开源, 免专利费的项目, 大大节省了我们的开发时间成本.
- WebRTC 由 Google 主导, 技术非常先进.
- 各大浏览器以及终端逐渐加大对 WebRTC 技术的支持.

WEBRTC OVERVIEW

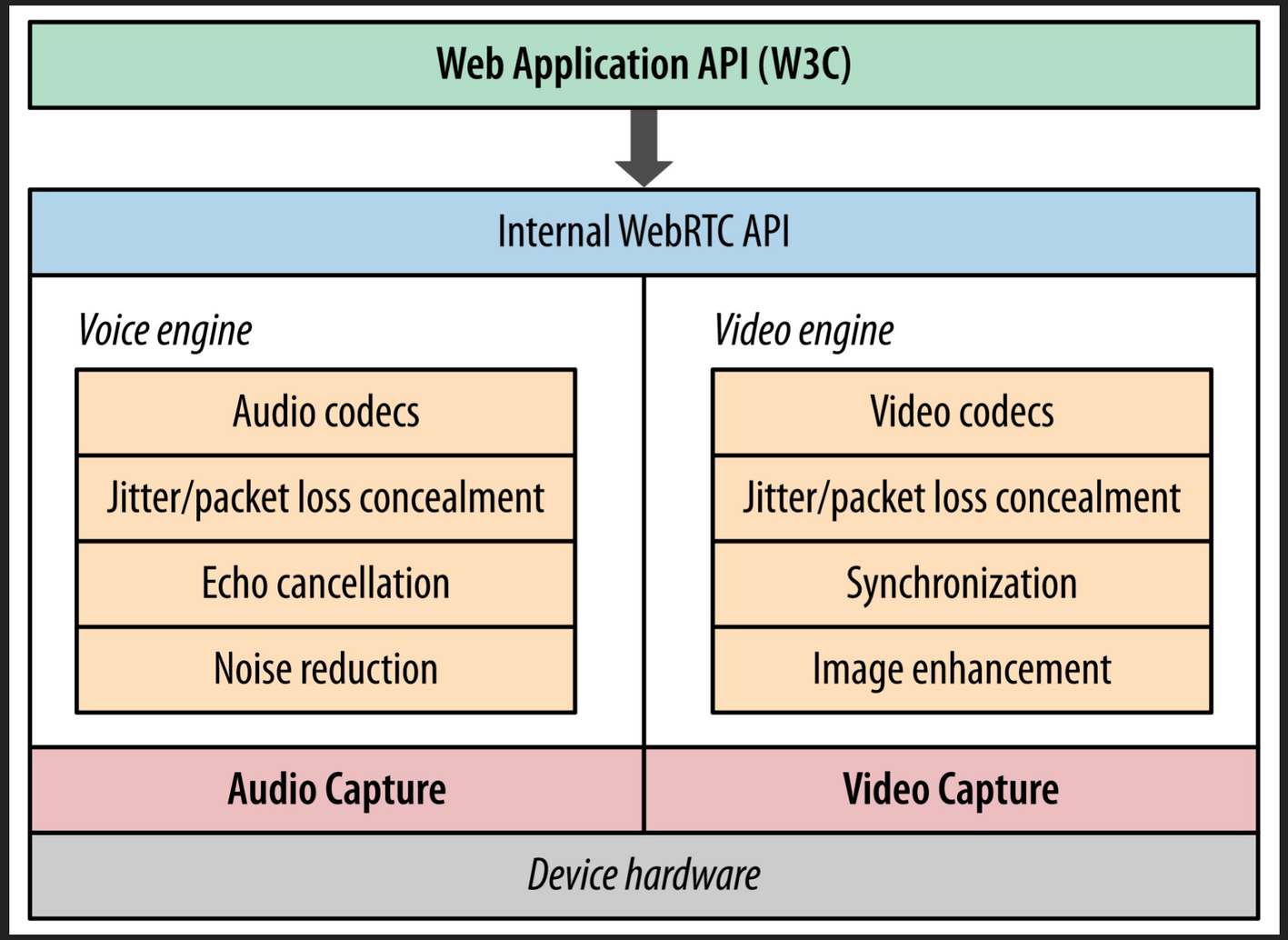
- WebRTC 全称是 Web Real-Time Communication.
- WebRTC 并不是一个单一的协议, 而是一个标准, 包含一堆协议和 API.
- WebRTC 通过提供简单易用的 JavaScript APIs 让浏览器拥有了 P2P 音视频和数据分享的能力, 同时不需要安装任何插件.

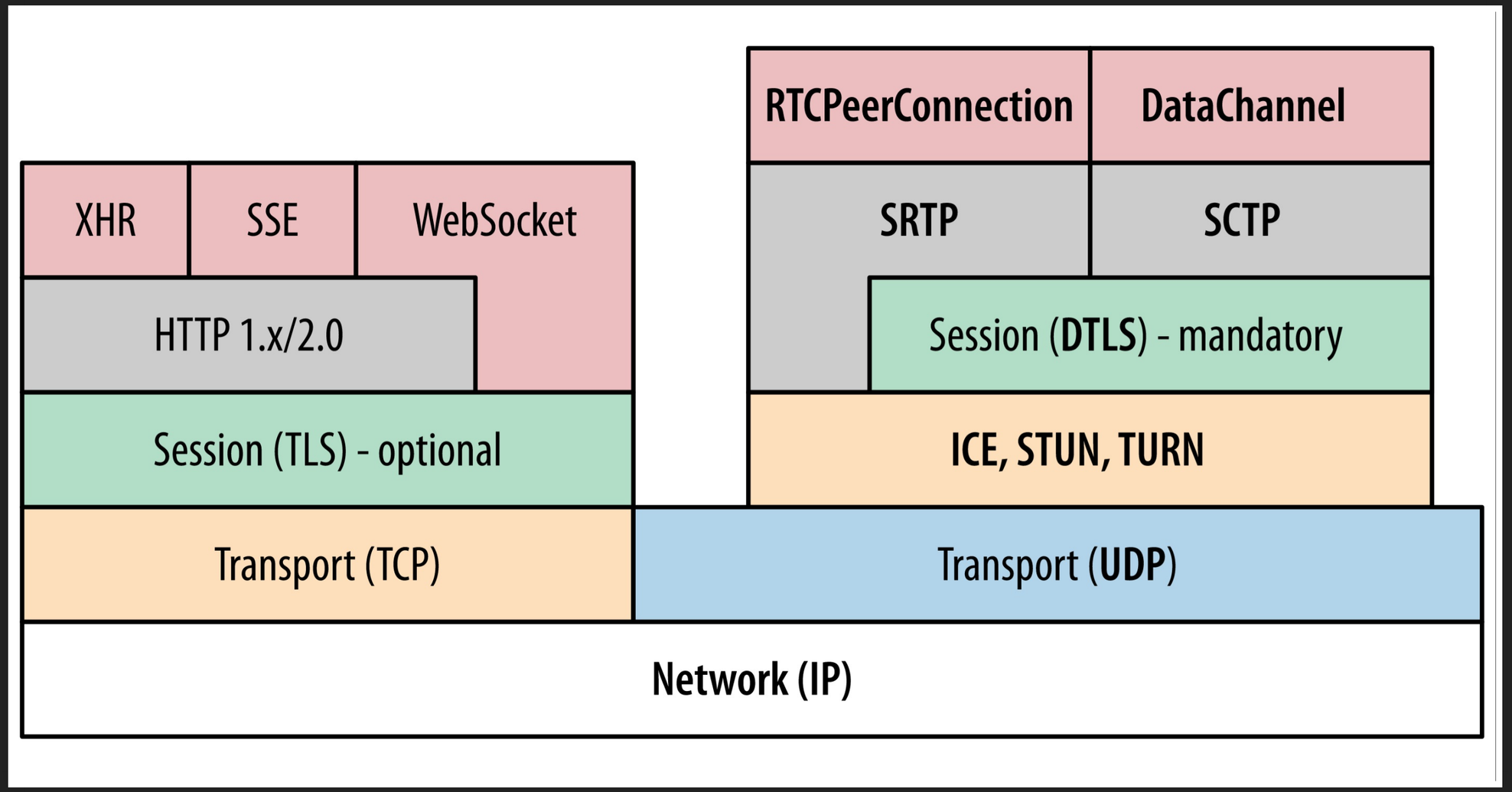
WEBRTC STANDARDS AND HISTORY

- 2010-05, Google 花了 $68.2 million 收购了 GIPS.
- 2011-05, Google 开源了 WebRTC 项目.
- WEBRTC W3C Working Group: 浏览器 API.
- RTCWEB IETF Working Group: 协议, 数据格式, 安全, P2P 等.
- WebRTC 并不是一个孤立的协议, 起初是为了浏览器与浏览器之间实时通信, 也可以通过信令协议对接现有的 SIP 客户端, PSTN 网络, 移动端等.

WEBRTC 的核心组成

- 音视频引擎. OPUS, VP8/VP9, H264
- 传输层协议. 底层传输协议为 UDP.
- 媒体协议. SRTP/SRTCP.
- 数据协议. DTLS/SCTP.
- P2P 内网穿透. STUN/TURN/ICE/Trickle ICE
- 信令与 SDP 协商. HTTP/WebSocket/SIP. Offer Answer 模型.


Web Application API (W3C)
Internal WebRTC API
Voice engine
Audio codecs
Jitter/packet loss concealment
Echo cancellation
Noise reduction
Video engine
Video codecs
Jitter/packet loss concealment
Synchronization
Image enhancement
Audio Capture
Video Capture
Device hardware


RTCPeerConnection
DataChannel
XHR
SSE
WebSocket
SRTP
SCTP
HTTP 1.x/2.0
Session (DTLS) - mandatory
Session (TLS) - optional
ICE, STUN, TURN
Transport (TCP)
Transport (UDP)
Network (IP)

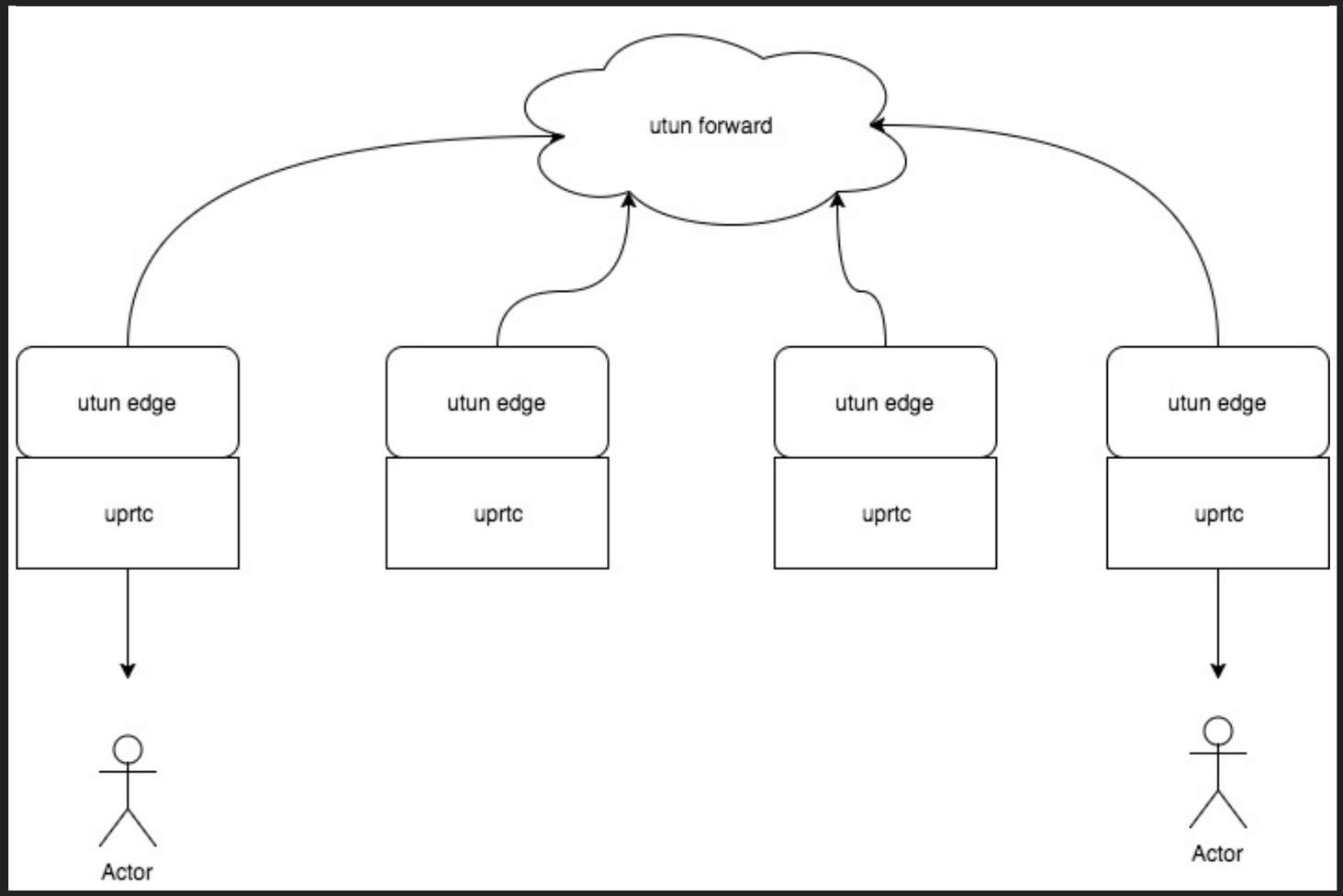
WEBRTC 标准文档

- W3C API 相关文档: https://github.com/w3c, MDN
- IETF 协议相关文档: https://datatracker.ietf.org/

我们的实时通信底层平台 UPRTC

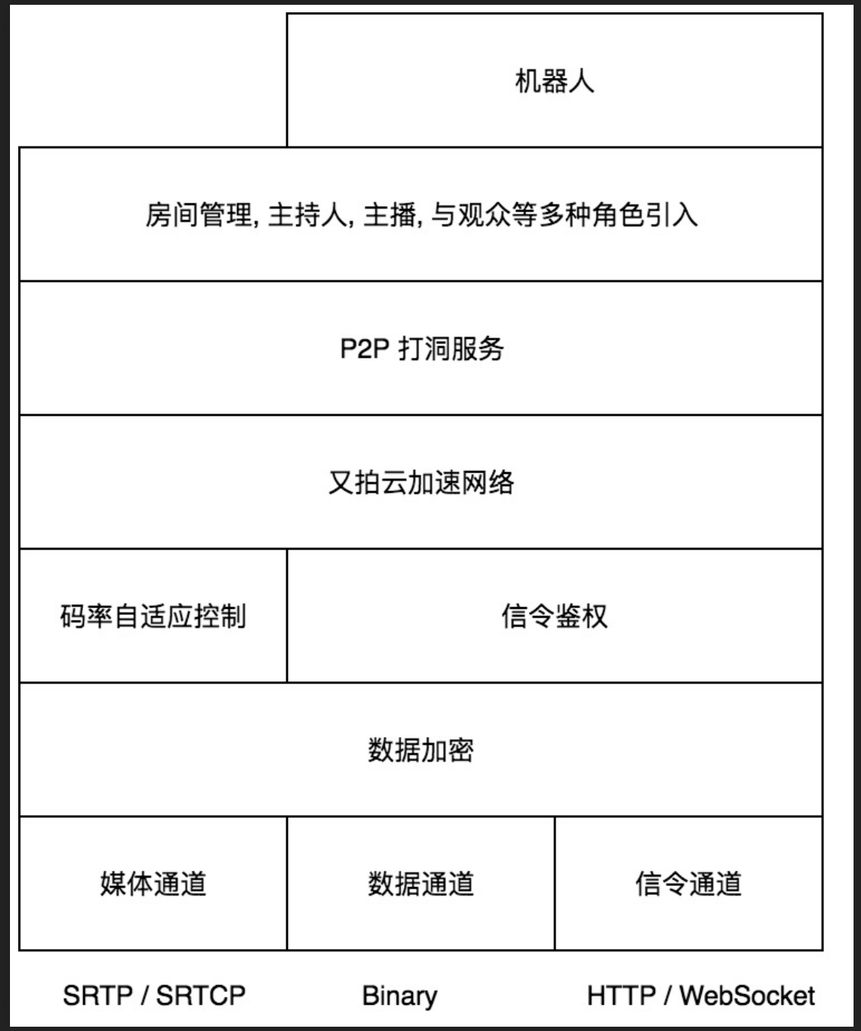
- 传统的 WebRTC 应用模式是 P2P 的, 我们改造成服务器中转的模式.
- 完全分布式系统, 部署到全国所有边缘节点, 通过我们的内部加速网络加速.
- 网络拥塞自适应控制, 较强的弱网适应能力.
- 针对底层开源组件进行优化改造, 支持高并发.
- 灵活高效的业务信令. 支持对敏感信令进行鉴权.

UPRTC 架构


utun forward
utun edge
uprtc
utun edge
uprtc
utun edge
uprtc
utun edge
uprtc
Actor
Actor

- utun 解决跨地区, 跨 ISP 延迟高且不稳定等网络问题.
- 覆盖 200 多个边缘节点, 4000 多台服务器.
- 覆盖 3 大运营商, 2 个小运营商.
- uprtc 实现媒体接入, 接入 Web 端与移动端.

UPRTC 技术组成


机器人
房间管理, 主持人, 主播, 与观众等多种角色引入
P2P 打洞服务
又拍云加速网络
码率自适应控制
信令鉴权
数据加密
媒体通道
数据通道
信令通道
SRTP / SRTCP
Binary
HTTP / WebSocket

基于我们底层 UPRTC 平台的第一个项目 - 上会

WEB 端

移动端

开发经验

- 我们选用的编码格式 H.264 + Opus.
- 修复 WebRTC 内核 iOS 端有音频处理过度消耗 CPU 的 BUG.
- 修复 WebRTC 的 core 音视频不同步的 BUG.
- Android 端 H.264 编码不支持高通以外芯片硬解码.
- 客户端解码能力有限, 总会话人数控制在 8 人以内.
- 服务器端需加入码率自适应算法, 自动根据参与人数控制总带宽在 2 Mbps以内.
- 美颜, 滤镜接入, 会增加处理延时, 所以对此性能要求非常高.

Q&A

Thanks

Contribute your ideas!