

毫无保留的直播技术分享

李智维

好处

- 直播是一种表现形式，未来可能存在各行各业，像图片文字一样流行，而直播技术就像金饭碗
- 直播研发工程师动辄 30k~50k
- 了解直播技术，也能帮助老板们、产品经理们在直播方面做出更好的产品

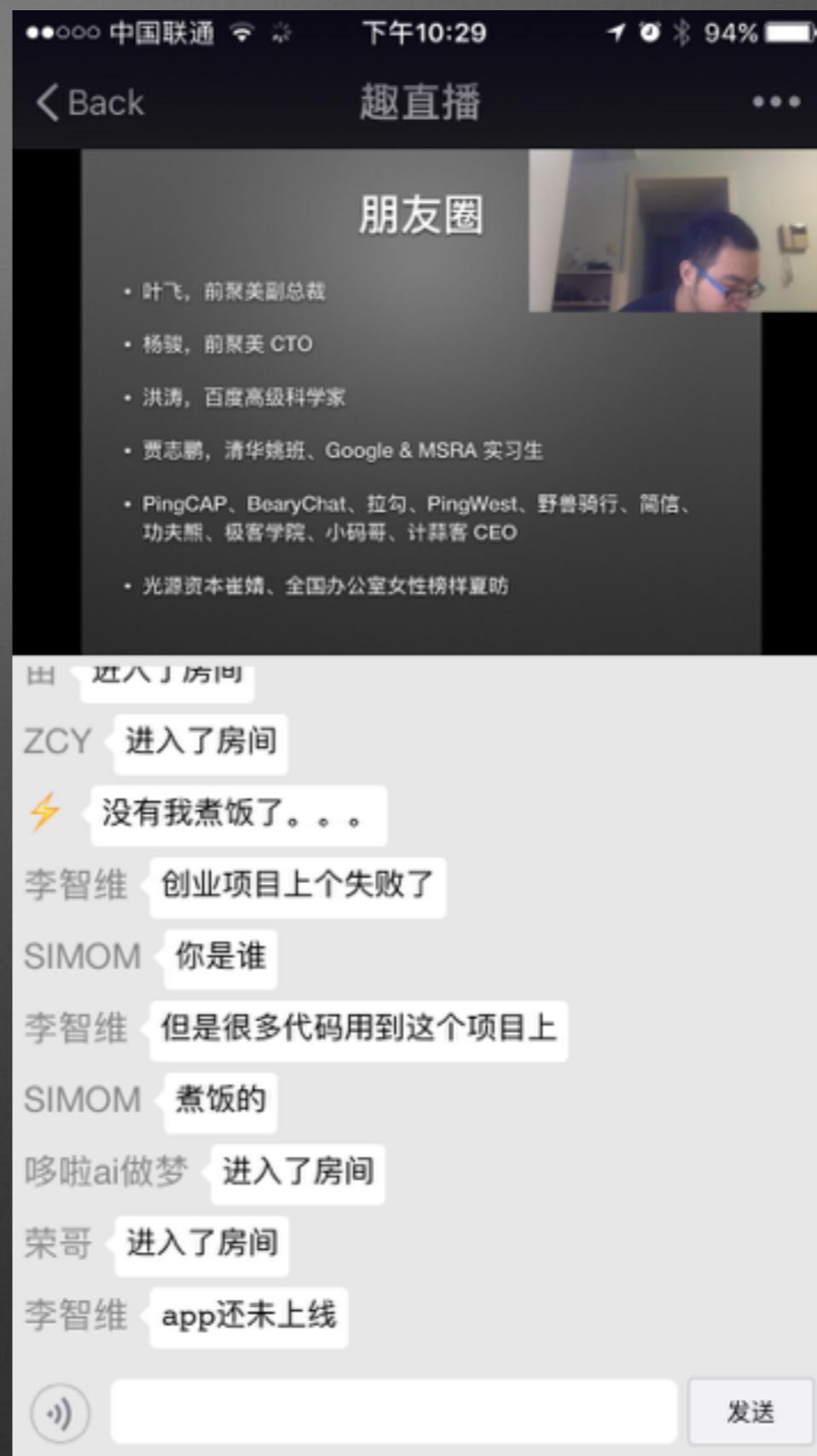
目前的成绩

- 完全自建的直播服务器、拥有所有直播相关的源码(站在开源项目的肩膀上)
- 同时在线 300 人观看的考验
- 服务器理论支持同时 2400 人观看(4台服务器，可水平扩展，也即10台服务器，便能支持 6000 人观看)
- 支持手机网页观看、桌面推流、iOS 观看、iOS 推流

讲座大纲

- 直播基础概念
- OBS 推流软件操作
- 服务端技术选型、搭建、集群
- 客户端 iOS 推流、播放
- 手机网页播放
- 直播费用、流量等介绍

趣直播-微信网页

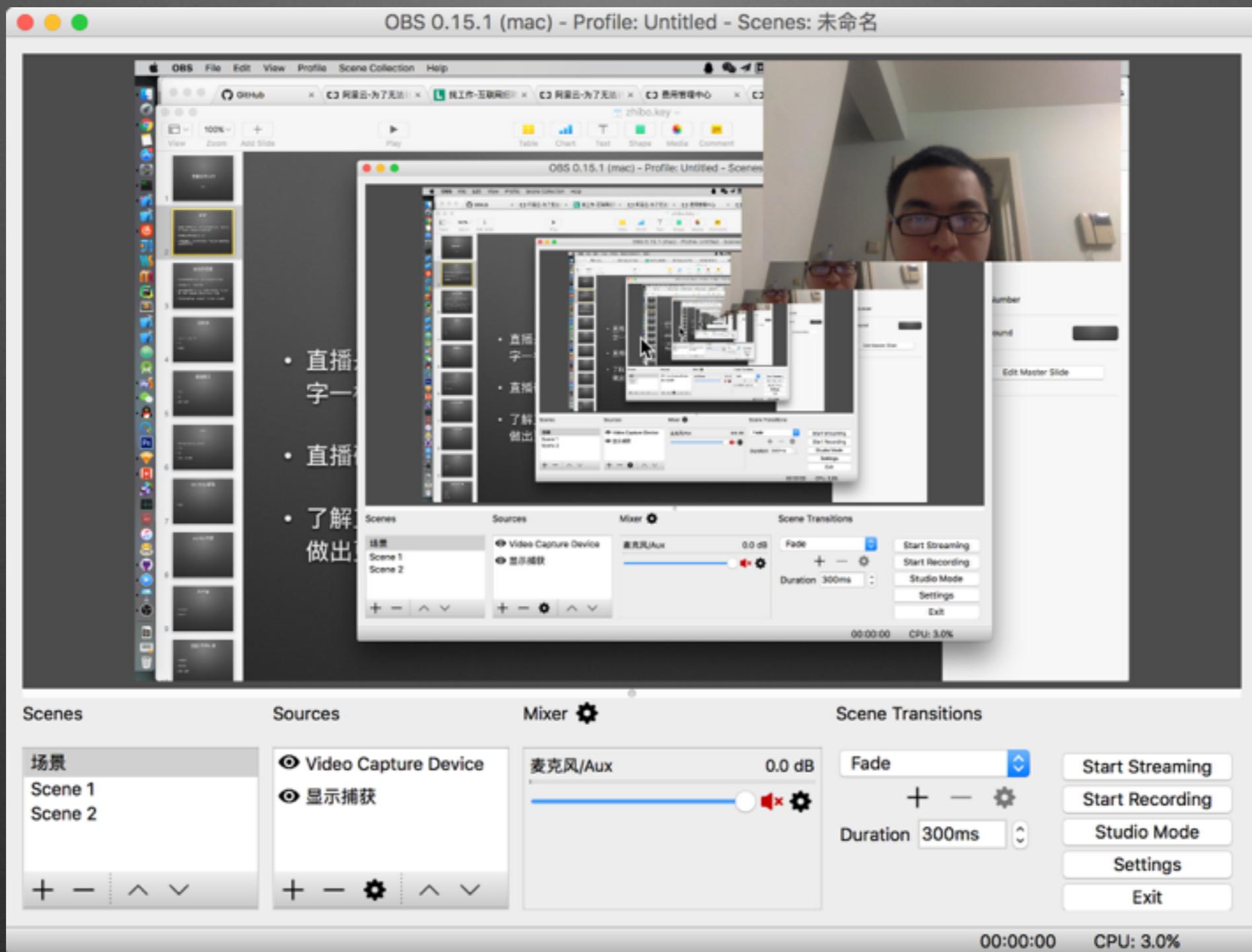


趣直播—iOS App(即将上线)

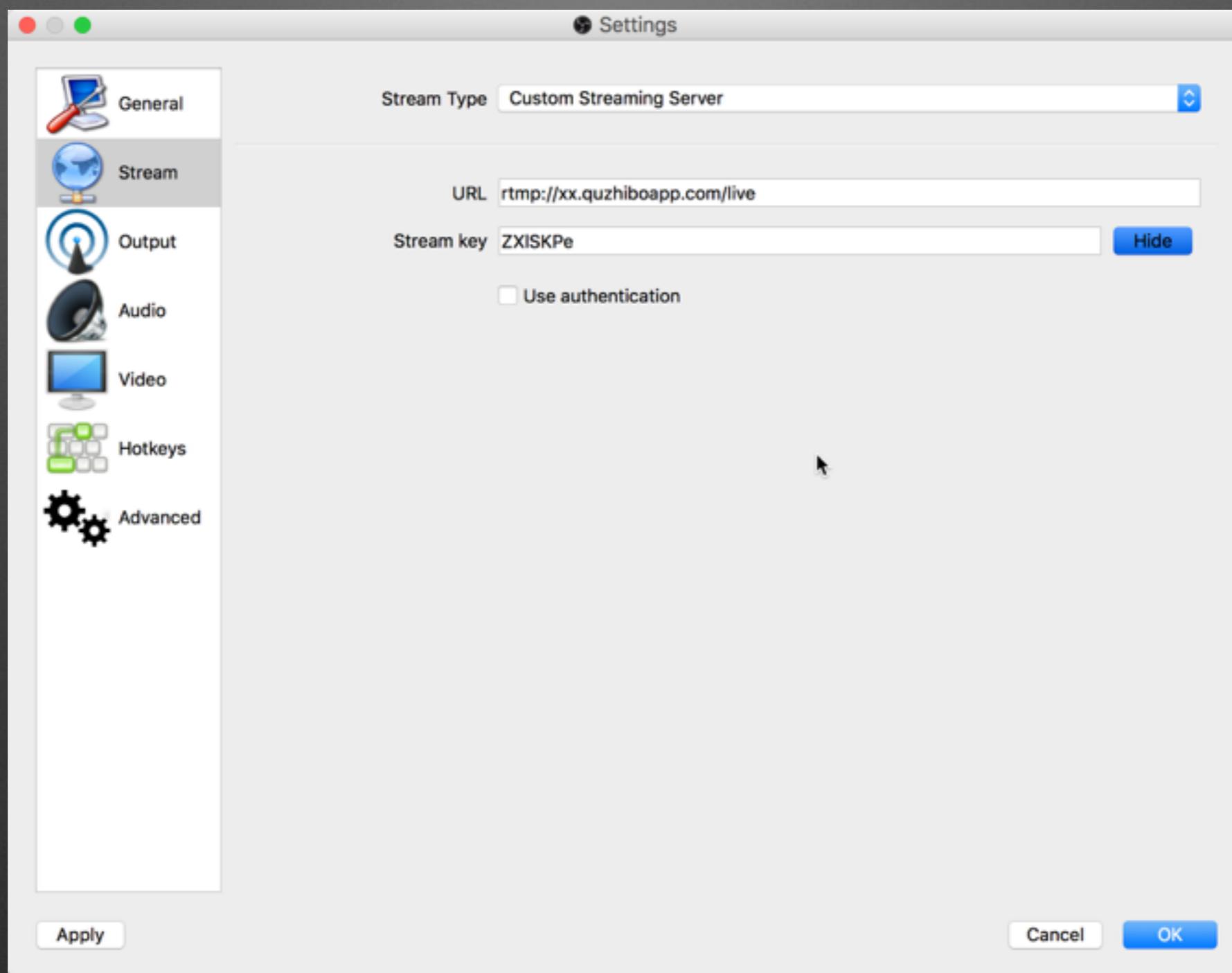


学会斗鱼直播
对直播有个直观认识

obs 推流软件



填写推流地址



填写地址

URL

Stream key

Use authentication



OBS-视频采集

- 屏幕
- 摄像头
- 文字
- 视频
- 图片

直播播放-VLC



Open Source

File Disc Network **Capture**

URL

To Open a usual network stream (HTTP, RTSP, RTMP, MMS, FTP, etc.), just enter the URL in the field above. If you want to open a RTP or UDP stream, press the button below.

Open RTP/UDP Stream

► Media Resource Locator (MRL)

Streaming/Saving: Settings...

Cancel Open

A screenshot of the VLC media player's "Network" interface. The URL field contains "rtmp://xx.quzhiboapp.com/live/test". The "Network" tab is selected. Below the URL field is a descriptive text about opening network streams. A button labeled "Open RTP/UDP Stream" is visible. At the bottom, there are buttons for "Cancel" and "Open". On the left, there are links for "Media Resource Locator (MRL)" and "Streaming/Saving" with a "Settings..." button. The overall interface is light gray with blue highlights for selected tabs and buttons.

直播播放-srs player

http://winlinvip.github.io/srs.release/trunk/research/players/srs_player.html

SRS SRS播放器 SRS编码器 SRS会议 SRS测网速 JWPlayer6播放器 AdobeOSMF播放器 VLC播放器

Usage: 输入RTMP/HTTP地址后点击“播放视频”即可播放视频

URL:

播放视频

注意：必须按照[SRS DEMO](#) 启动和设置服务器，下面所有的链接才能观看。

[1] SRS示例播放流: 原始流

[2] SRS示例播放流: 原始流HLS

[3] SRS示例播放流: 转码配置LD流

进阶-ffmpeg 推流

```
ffmpeg -re -i test.flv \
-vcodec copy -acodec copy \
-f flv -y rtmp://192.168.1.170/live/livestream; \
```

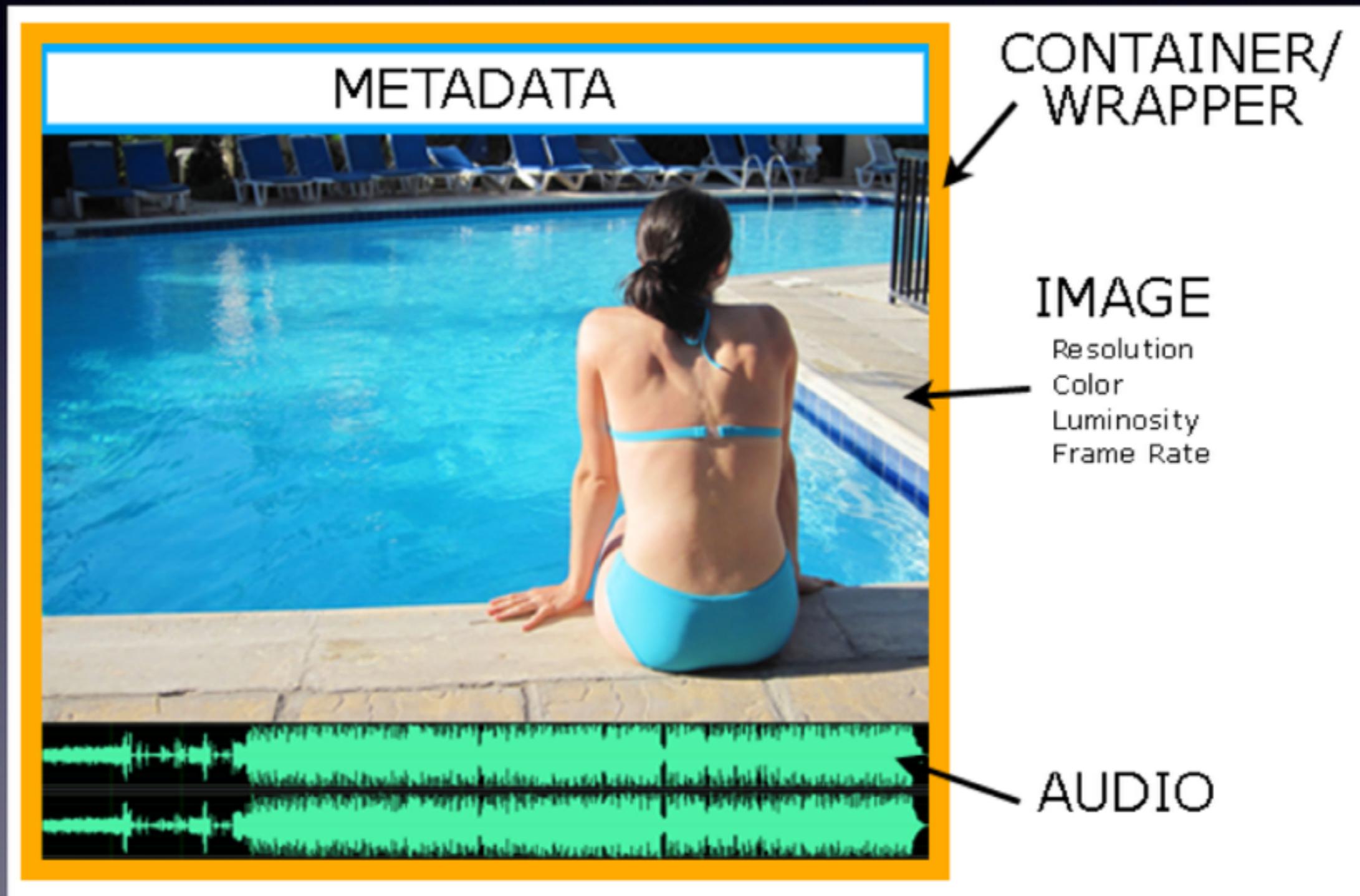
1)指定视频源 2)指定推流地址

直播基础概念

直播基础概念

- 视频
- rtmp
- hls
- 推流、拉流

理性认知视频



视频相关动词

术语	典型场景	典型用例
解码(<i>Decoding</i>)	点播、回放(<i>Playback</i>)	源文件 -> <i>Decoding</i> -> 播放
编码(<i>Encoding</i>)	直播(<i>Streaming</i>) -> 录像(<i>Video</i>)	直播流 -> <i>Encoding</i> -> 视频录像
转码(<i>Transcoding</i>)	特效加工处理 文件格式转换	源文件 -> <i>Decoding</i> -> 加工 -> <i>Encoding</i> -> 目标文件

模型

通用直播模型



iOS、Android 用Rtmp；手机网页用 HLS；桌面网页用 FLV；

RTMP

- 一个流媒体传输协议(Real Time Message Protocol), 来自Adobe
- 定义了视频、音频怎么实时传输
- 每个时刻的数据, 收到立刻转发
- 建立在长连接 TCP 上
- TCP 上的一层协议
- 延时 1~3 秒
- 地址行如 rtmp://xx.quzhiboapp.com/live/test

HLS

- HLS 全称 HTTP Live Streaming
- 短连接 HTTP
- 10秒切割一段直播，60秒一个播放列表，客户端隔一段时间请求播放列表
- 延时 > 10 秒

HTTP-FLV

- <http://ossrs.net:8081/live/livestream.flv>
- 以FLV 的方式封装 RTMP 流，形式上稍微转换了一下
- 有RTMP的所有特征，譬如集群、低延迟、热备、GOP cache
- 而且有HTTP的优势，譬如302、穿墙、通用
- 延时 1~3 秒

Why HTTP FLV

- 低延迟，和 RTMP 一样
- 穿墙，很多防火墙会墙掉 rtmp，但 http 不会
- 容错，SRS的HTTP FLV回源时可以回多个，支持多级热备
- 通用，Flash可以播RTMP，也可以播HTTP FLV，App 也支持
- 简单，FLV是最简单的流媒体封装，HTTP是最广泛的协议

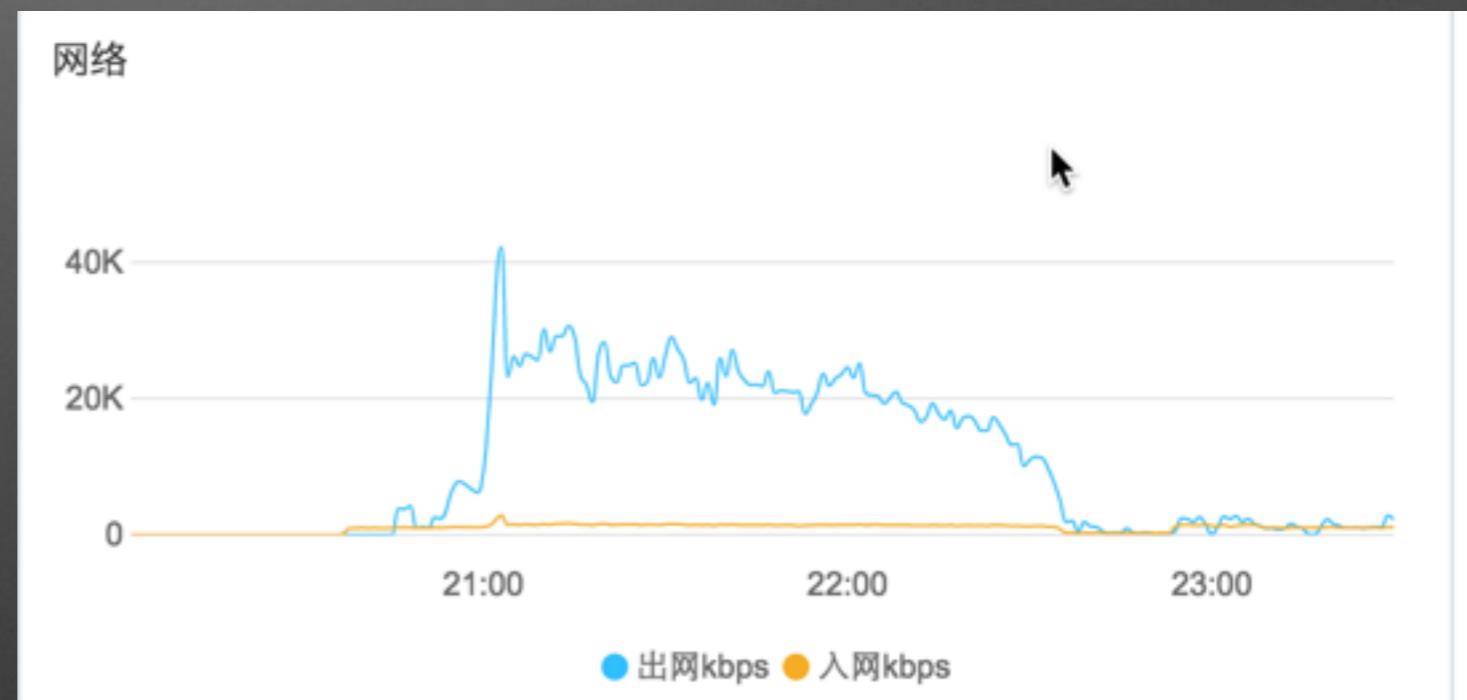
协议差异

	全称	协议	原理	延时
RTMP	Real Time Messaging Protocol	长连接 TCP	每个时刻的数据，收到后立刻转发	1~3 秒
HLS	HTTP Live Streaming	短连接 HTTP	集合一段时间数据，生成 ts 切片文件，更新m3u8	> 10 秒
HTTP-FLV	RTMP over HTTP	长连接 HTTP	同RTMP，使用 HTTP协议	1~3 秒



服务端

- nginx-rtmp-module
- srs
- 阿里云



srs

- <https://github.com/ossrs/srs>
- rtmp
- hls
- rtmp、hls 集群

srs 编译

```
./configure --disable-all --with-ssl  
&& make
```

conf 配置

```
# conf/rtmp.conf
listen          1935;
max_connections 1000;
vhost _defaultVhost_ {
}
```

启动 srs

```
./objs/srs -c conf/rtmp.conf
```

ffmpeg 推流

```
for((:)); do \
    ./objs/ffmpeg/bin/ffmpeg -re -i ./doc/source.
200kbps.768x320.flv \
    -vcodec copy -acodec copy \
    -f flv -y rtmp://192.168.1.170/live/livestream; \
    sleep 1; \
done
```

rtmp play

rtmp://192.168.1.170:1935/live/livestream

srs 转码

```
19vhost xx.quzhiboapp.com {  
20  
21    transcode {  
22        enabled      on;  
23        ffmpeg       ./objs/ffmpeg/bin/ffmpeg;  
24        engine ff {  
25            enabled      on;  
26            vcodec       libx264;  
27            vbitrate     500;  
28            vfps         20;  
29            vwidth        400;  
30            vthreads      12;  
31            vprofile      main;  
32            vpreset       medium;  
33            acodec        copy;  
34            #acodec      libaacplus;    ↗  
35            #abitrage     70;  
36            #asample_rate 44100;  
37            #achannels    2;  
38            output        rtmp://127.0.0.1:[port]/[app]?vhost=__defaultVhost__[s  
39        }  
40    }  
41}
```

srs hls 配置

客户端

- VideoCore, 推流用, 底层 ffmpeg。
- Ijkplayer 播放用, 播直播用, 底层用 ffmpeg play。

iOS VideoCore 模块解析

- Rtmp Session 等 Rtmp 管理类
- VCSimpleSession 采集管理类
- Video Mixer、Audio Mixer
- BeautyVideoFilter 等美颜Filter
- VCPreviewView 视频预览
- ffmpeg

- <https://github.com/runner365/LiveVideoCoreSDK>

VCSimpleSession

- 核心采集管理类
- 核心函数 `setupGraph`
- 初始化采集视频，输出到 `VideoMixer` 上
- 初始化采集音频，输出到 `AudioMixer` 上
- 将当前采集的视频输出渲染到 `VCPreviewView` 上

AudioMixer

- 支持多个音频源，比如主播的声音和背景音乐
- 视频源同理

VCPreviewView

- 把摄像头采集的数据直接渲染到屏幕上

移动Web

```
<video id="player1" width="100%" height="100%" preload="preload" controls webkit-playsinline :src="live.hlsUrl"></video>
```

- Video 标签, video 标签的事件
- hls 播放
- inline 播放的问题
- 播放预告的做法
- 聊天用 LeanCloud IM SDK

服务器集群

- 1) cheer->live1->live2->south , 一个一个推流, forward
- 2) 推送 rtmp 流, 在每个服务器进行 hls 切割
- 3) 随机分配一台服务器, 暂时没负载
- 4) cheer 服务器转码, 之后进行转码后的 rtmp 推流