



作者 峥吖 (/users/b09c3959ab3b) 2016.08.25 20:29\*

写了36307字，被6256人关注，获得了2592个喜欢  
(/users/b09c3959ab3b)

✓ 正在关注 (/users/b09c3959ab3b/toggle\_like)

# 【如何快速的开发一个完整的iOS直播app】(原理篇)

字数1481 阅读4635 评论41 喜欢217

## 前言

大半年没写博客了，但我一直关注着互联网的动向，最近会研究很多东西，并分享，今年移动直播行业的兴起，诞生了一大批网红，甚至明星也开始直播了，因此不得不跟上时代的步伐，由于第一次接触的原因，因此花了很多时间了解直播，整理了直播的原理，当前只是 原理篇 ，后续会持续发布 实战篇 ，教你 从零开始搭建一个完整的iOS直播app ，希望能帮助到更多的人更快的了解直播。

如果喜欢我的文章，可以关注我微博: 咻了个峥 (<http://weibo.com/2034818060/profile?rightmod=1&wvr=6&mod=personinfo>), 也可以来小码哥 (<http://www.520it.com>)，了解下我们的iOS 培训课程。后续还会更新更多内容, 有任何问题，欢迎简书留言 峥吖 ([http://www.jianshu.com/users/b09c3959ab3b/latest\\_articles](http://www.jianshu.com/users/b09c3959ab3b/latest_articles))。。。

## 一、个人见解（直播难与易）

**直播难**：个人认为要想把直播从零开始做出来，绝对是牛逼中的牛逼，大牛中的大牛，因为直播中运用到的技术难点非常之多，视频/音频处理，图形处理，视频/音频压缩，CDN分发，即时通讯等技术，每一个技术都够你学几年的。

**直播易**：已经有各个领域的大牛，封装好了许多牛逼的框架，我们只需要用别人写好的框架，就能快速的搭建一个直播app，也就是传说中的站在大牛肩膀上编程。

## 二、了解直播

### 热门直播产品

映客，斗鱼，熊猫，虎牙，花椒等等

### 直播效果图





直播效果.jpeg

## 1.一个完整直播app功能(来自落影loyinglin (<http://www.jianshu.com/p/d99e83cab39a>)分享)

- 1、 聊天
  - 私聊、聊天室、点亮、推送、黑名单等;
- 2、 礼物
  - 普通礼物、豪华礼物、红包、排行榜、第三方充值、内购、礼物动态更新、提现等 ;
- 3、 直播列表
  - 关注、热门、最新、分类直播用户列表等 ;

- 4、 自己直播
  - 录制、推流、解码、播放、美颜、心跳、后台切换、主播对管理员操作、管理员对用户等；
- 5、 房间逻辑
  - 创建房间、进入房间、退出房间、关闭房间、切换房间、房间管理员设置、房间用户列表等；
- 6、 用户逻辑
  - 普通登陆、第三方登陆、注册、搜索、修改个人信息、关注列表、粉丝列表、忘记密码、查看个人信息、收入榜、关注和取关、检索等；
- 7、 观看直播
  - 聊天信息、滚屏弹幕、礼物显示、加载界面等；
- 8、 统计
  - APP业务统计、第三方统计等；
- 9、 超管
  - 禁播、隐藏、审核等；

## 2.一个完整直播app原理

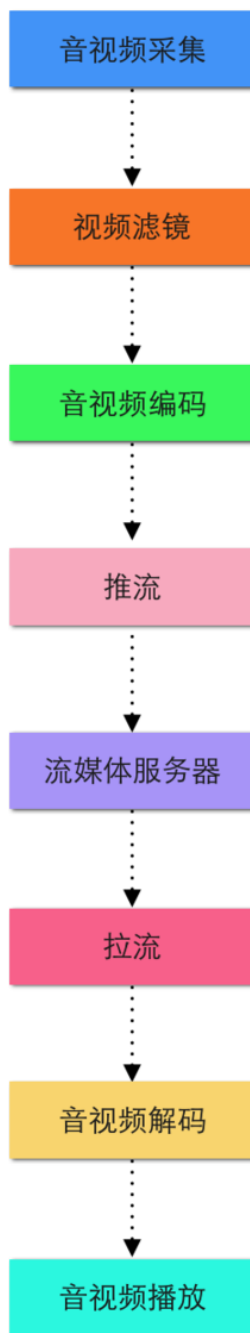
直播原理：把主播录制的视频，推送到服务器，在由服务器分发给观众观看。

直播环节：推流端（采集、美颜处理、编码、推流）、服务端处理（转码、录制、截图、鉴黄）、播放器（拉流、解码、渲染）、互动系统（聊天室、礼物系统、赞）

## 3.一个完整直播app实现流程

1.采集、2.滤镜处理、3.编码、4.推流、5.CDN分发、6.拉流、7.解码、8.播放、9.聊天互动

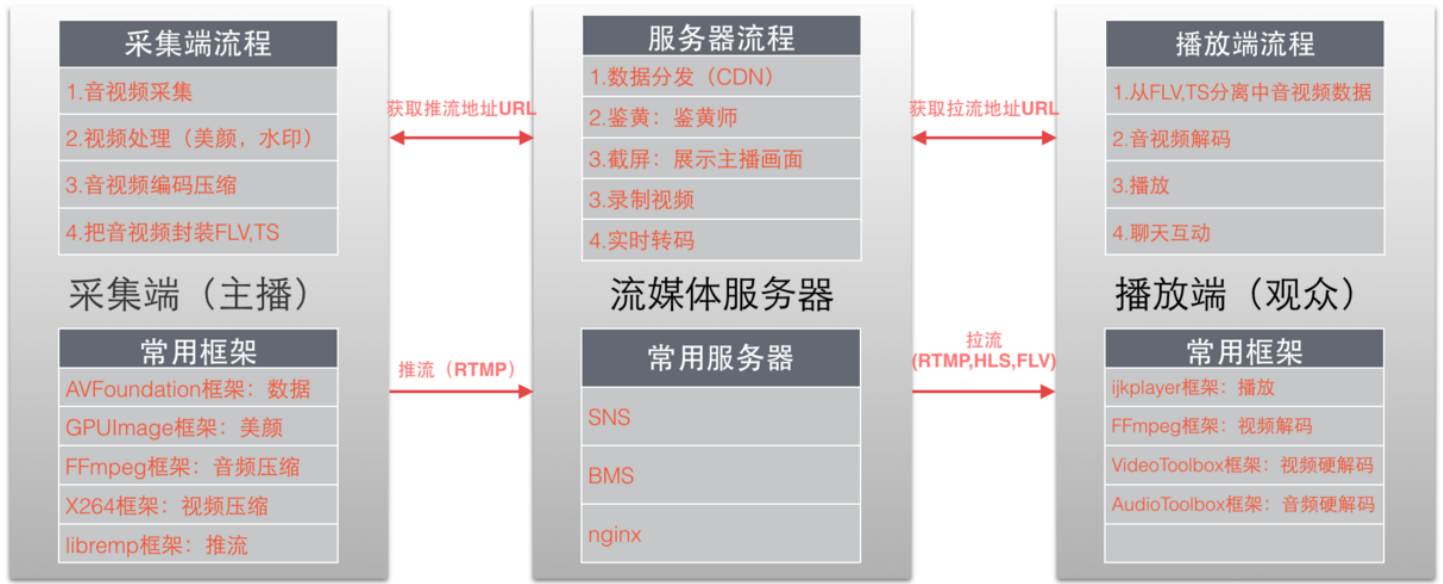
## 直播App流程



直播流程.png

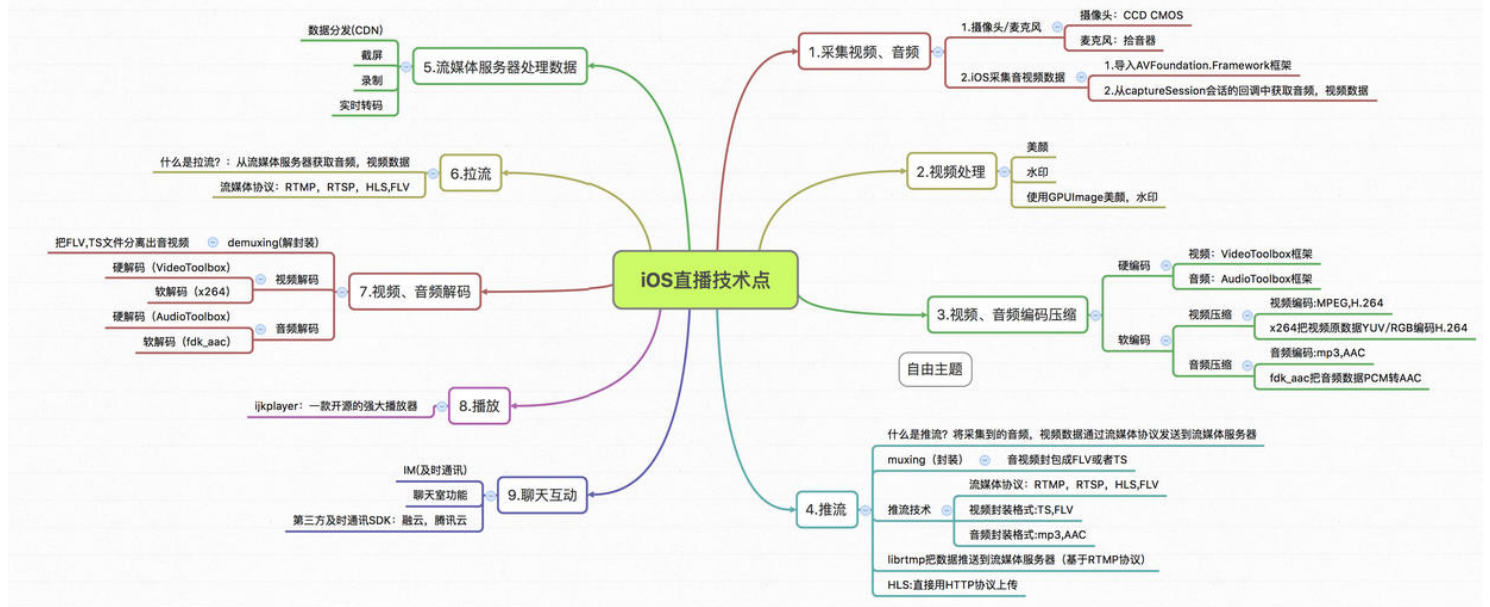
### 4. 一个完整直播app架构

# 直播App架构



直播架构.png

## 5. 一个完整直播app技术点



WeChat\_1472043345.jpeg

## 三、了解流媒体（直播需要用到流媒体）

- 流媒体开发：网络层(socket或st)负责传输，协议层(rtmp或hls)负责网络打包，封装层(flv、ts)负责编解码数据的封装，编码层(h.264和aac)负责图像，音频压缩。
- 帧：每帧代表一幅静止的图像

- **GOP** : ( Group of Pictures ) 画面组 , 一个GOP就是一组连续的画面 , 每个画面都是一帧 , 一个GOP就是很多帧的集合
  - 直播的数据 , 其实是一组图片 , 包括I帧、P帧、B帧 , 当用户第一次观看的时候 , 会寻找I帧 , 而播放器会到服务器寻找到最近的I帧反馈给用户。因此 , GOP Cache增加了端到端延迟 , 因为它必须要拿到最近的I帧
  - GOP Cache的长度越长 , 画面质量越好
- **码率** : 图片进行压缩后每秒显示的数据量。
- **帧率** : 每秒显示的图片数。影响画面流畅度 , 与画面流畅度成正比 : 帧率越大 , 画面越流畅 ; 帧率越小 , 画面越有跳动感。
  - 由于人类眼睛的特殊生理结构 , 如果所看画面之帧率高于16的时候 , 就会认为是连贯的 , 此现象称之为视觉暂留。并且当帧速达到一定数值后 , 再增长的话 , 人眼也不容易察觉到有明显的流畅度提升了。
- **分辨率** : (矩形)图片的长度和宽度 , 即图片的尺寸
- **压缩前的每秒数据量** : 帧率X分辨率(单位应该是若干个字节)
- **压缩比** : 压缩前的每秒数据量/码率 ( 对于同一个视频源并采用同一种视频编码算法 , 则 : 压缩比越高 , 画面质量越差。 )
- **视频文件格式** : 文件的后缀 , 比如 .wmv , .mov , .mp4 , .mp3 , .avi ,
  - 主要用处 , 根据文件格式 , 系统会自动判断用什么软件打开 ,  
注意: 随意修改文件格式 , 对文件的本身不会造成太大的影响 , 比如把avi改成mp4,文件还是avi.
- **视频封装格式** : 一种储存视频信息的容器 , 流式封装可以有 TS、FLV 等 , 索引式的封装有 MP4,MOV,AVI 等 ,
  - 主要作用 : 一个视频文件往往会包含图像和音频 , 还有一些配置信息(如图像和音频的关联 , 如何解码它们等) : 这些内容需要按照一定的规则组织、封装起来.
  - 注意 : 会发现封装格式跟文件格式一样 , 因为一般视频文件格式的后缀名即采用相应的视频封装格式的名称,所以视频文件格式就是视频封装格式。
- **视频封装格式和视频压缩编码标准** : 就好像项目工程和编程语言 , 封装格式就是一个项目的工程 , 视

频编码方式就是编程语言，一个项目工程可以用不同语言开发。

## 四、直播基础知识介绍：

### 1.采集视频、音频

#### \* 1.1 采集视频、音频编码框架 \*

- **AVFoundation**:AVFoundation是用来播放和创建实时的视听媒体数据的框架，同时提供Objective-C接口来操作这些视听数据，比如编辑，旋转，重编码

#### \* 1.2 视频、音频硬件设备 \*

- **CCD**:图像传感器：用于图像采集和处理的过程，把图像转换成电信号。
- **拾音器**:声音传感器：用于声音采集和处理的过程，把声音转换成电信号。
- **音频采样数据**:一般都是PCM格式
- **视频采样数据**:一般都是YUV,或RGB格式，采集到的原始音视频的体积是非常大的，需要经过压缩技术处理来提高传输效率

### 2.视频处理（美颜，水印）

- **视频处理原理**:因为视频最终也是通过GPU，一帧一帧渲染到屏幕上的，所以我们可以利用OpenGL ES，对视频帧进行各种加工，从而视频各种不同的效果，就好像一个水龙头流出的水，经过若干节管道，然后流向不同的目标
  - 现在的各种美颜和视频添加特效的app都是利用GPUImage这个框架实现的,.

#### \* 视频处理框架 \*

- **GPUImage**:GPUImage是一个基于OpenGL ES的一个强大的图像/视频处理框架,封装好了各种滤镜同时也可以编写自定义的滤镜,其本身内置了多达120多种常见的滤镜效果。
- **OpenGL**:OpenGL（全写Open Graphics Library）是个定义了一个跨编程语言、跨平台的编程接口的规格，它用于三维图象（二维的亦可）。OpenGL是个专业的图形程序接口，是一个功能强大，调用方便的底层图形库。
- **OpenGL ES**:OpenGL ES (OpenGL for Embedded Systems) 是 OpenGL三维图形 API 的子集，针对手机、PDA和游戏主机等嵌入式设备而设计。

### 3.视频编码解码

### \* 3.1 视频编码框架 \*

- `FFmpeg` :是一个跨平台的开源视频框架,能实现如视频编码,解码,转码,串流,播放等丰富的功能。其支持的视频格式以及播放协议非常丰富,几乎包含了所有音视频编解码、封装格式以及播放协议。
  - `-Libswresample`:可以对音频进行重采样,rematrixing 以及转换采样格式等操作。
  - `-Libavcodec`:提供了一个通用的编解码框架,包含了许多视频,音频,字幕流 等编码/解码器。
  - `-Libavformat`:用于对视频进行封装/解封装。
  - `-Libavutil`:包含一些共用的函数,如随机数生成,数据结构,数学运算等。
  - `-Libpostproc`:用于进行视频的一些后期处理。
  - `-Libswscale`:用于视频图像缩放,颜色空间转换等。
  - `-Libavfilter`:提供滤镜功能。
- `x264` :把视频原数据YUV编码压缩成H.264格式
- `VideoToolbox` :苹果自带的视频硬解码和硬编码API ,但是在iOS8之后才开放。
- `AudioToolbox` :苹果自带的音频硬解码和硬编码API

### \* 3.2 视频编码技术 \*

- 视频压缩编码标准 : 对视频进行压缩(视频编码)或者解压缩(视频解码)的编码技术 ,比如 `MPEG` , `H.264` ,这些视频编码技术是压缩编码视频的
  - 主要作用 :是将视频像素数据压缩成为视频码流 ,从而降低视频的数据量。如果视频不经过压缩编码的话 ,体积通常是非常大的 ,一部电影可能就要上百G的空间。
  - 注意 :最影响视频质量的是其视频编码数据和音频编码数据 ,跟封装格式没有多大关系
- `MPEG` :一种视频压缩方式 ,它采用了帧间压缩 ,仅存储连续帧之间有差别的地方 ,从而达到较大的压缩比
- `H.264/AVC` :一种视频压缩方式,采用事先预测和与MPEG中的P-B帧一样的帧预测方法压缩 ,它可以根据需要产生适合网络情况传输的视频流,还有更高的压缩比 ,有更好的图象质量
  - 注意1 :如果是从单个画面清晰度比较 ,MPEG4有优势 ;从动作连贯性上的清晰度 ,H.264有优势
  - 注意2 :由于264的算法更加复杂 ,程序实现烦琐 ,运行它需要更多的处理器和内存资源。因此 ,运行264对系统要求是比较高的。



- 注意3 :由于264的实现更加灵活，它把一些实现留给了厂商自己去实现，虽然这样给实现带来了许多好处，但是不同产品之间互通成了很大的问题，造成了通过A公司的编码器编出的数据，必须通过A公司的解码器去解这样尴尬的事情
- H.265/HEVC :一种视频压缩方式,基于H.264，保留原来的某些技术，同时对一些相关的技术加以改进，以改善码流、编码质量、延时和算法复杂度之间的关系，达到最优化设置。
  - H.265 是一种更为高效的编码标准，能够在同等画质效果下将内容的体积压缩得更小，传输时更快更省带宽
  - I帧 :(关键帧)保留一副完整的画面，解码时只需要本帧数据就可以完成（因为包含完整画面）
- P帧 :(差别帧)保留这一帧跟之前帧的差别，解码时需要用之前缓存的画面叠加上本帧定义的差别，生成最终画面。（P帧没有完整画面数据，只有与前一帧的画面差别的数据）
- B帧 :(双向差别帧)保留的是本帧与前后帧的差别，解码B帧，不仅要取得之前的缓存画面，还要解码之后的画面，通过前后画面的与本帧数据的叠加取得最终的画面。B帧压缩率高，但是解码时CPU会比较累
- 帧内 ( Intraframe ) 压缩 :当压缩一帧图像时，仅考虑本帧的数据而不考虑相邻帧之间的冗余信息,帧内一般采用有损压缩算法
- 帧间 ( Interframe ) 压缩 :时间压缩 ( Temporal compression )，它通过比较时间轴上不同帧之间的数据进行压缩。帧间压缩一般是无损的
- muxing (合成) :将视频流、音频流甚至是字幕流封装到一个文件中( 容器格式 ( FLV , TS ) )，作为一个信号进行传输。

### \* 3.3 音频编码技术 \*

- AAC , mp3 : 这些属于音频编码技术,压缩音频用

### \* 3.4码率控制 \*

- 多码率 :观众所处的网络情况是非常复杂的，有可能是WiFi，有可能4G、3G、甚至2G，那么怎么满足多方需求呢？多搞几条线路，根据当前网络环境自定义码率。
  - 列如：常常看见视频播放软件中的1024，720，高清，标清，流畅等，指的就是各种码率。

### \* 3.5 视频封装格式 \*

- **TS** : 一种流媒体封装格式, 流媒体封装有一个好处, 就是不需要加载索引再播放, 大大减少了首次载入的延迟, 如果片子比较长, mp4文件的索引相当大, 影响用户体验
  - 为什么要用TS : 这是因为两个TS片段可以无缝拼接, 播放器能连续播放
- **FLV** : 一种流媒体封装格式, 由于它形成的文件极小、加载速度极快, 使得网络观看视频文件成为可能, 因此FLV格式成为了当今主流视频格式

## 4.推流

### \* 4.1 数据传输框架 \*

**librtmp** : 用来传输RTMP协议格式的数据

### \* 4.2 流媒体数据传输协议 \*

- **RTMP** : 实时消息传输协议, Adobe Systems公司为Flash播放器和服务器之间音频、视频和数据传输开发的开放协议, 因为是开放协议所以都可以使用了。
  - RTMP协议用于对象、视频、音频的传输。
  - 这个协议建立在TCP协议或者轮询HTTP协议之上。
  - RTMP协议就像一个用来装数据包的容器, 这些数据可以是FLV中的视音频数据。一个单一的连接可以通过不同的通道传输多路网络流, 这些通道中的包都是按照固定大小的包传输的

**chunk** : 消息包

## 5.流媒体服务器

### \* 5.1常用服务器 \*

- **SRS** : 一款国人开发的优秀开源流媒体服务器系统
- **BMS** : 也是一款流媒体服务器系统, 但不开源, 是SRS的商业版, 比SRS功能更多
- **nginx** : 免费开源web服务器, 常用来配置流媒体服务器。

### \* 5.2数据分发 \*

- **CDN** : (Content Delivery Network), 即内容分发网络, 将网站的内容发布到最接近用户的网络”边

缘”，使用户可以就近取得所需的内容，解决 Internet网络拥挤的状况，提高用户访问网站的响应速度.

- CDN ：代理服务器，相当于一个中介。
- CDN工作原理 ：比如请求流媒体数据
  - 1.上传流媒体数据到服务器（源站）
  - 2.源站存储流媒体数据
  - 3.客户端播放流媒体，向CDN请求编码后的流媒体数据
  - 4.CDN的服务器响应请求，若节点上没有该流媒体数据存在，则向源站继续请求流媒体数据；若节点上已经缓存了该视频文件，则跳到第6步。
  - 5.源站响应CDN的请求，将流媒体分发到相应的CDN节点上
  - 6.CDN将流媒体数据发送到客户端
- 回源 ：当有用户访问某一个URL的时候，如果被解析到的那个CDN节点没有缓存响应的内容，或者是缓存已经到期，就会回 源站 去获取搜索。如果没有人访问，那么CDN节点不会主动去 源站 拿.
- 带宽 :在固定的时间可传输的数据总量，
  - 比如64位、800MHz的前端总线，它的数据传输率就等于 $64\text{bit} \times 800\text{MHz} \div 8(\text{Byte}) = 6.4\text{GB/s}$
- 负载均衡：由多台服务器以对称的方式组成一个服务器集合，每台服务器都具有等价的地位，都可以单独对外提供服务而无须其他服务器的辅助。
  - 通过某种负载分担技术，将外部发送来的请求均匀分配到对称结构中的某一台服务器上，而接收到请求的服务器独立地回应客户的请求。
  - 均衡负载能够平均分配客户请求到服务器列阵，籍此提供快速获取重要数据，解决大量并发访问服务问题。
  - 这种群集技术可以用最少的投资获得接近于大型主机的性能。
- QoS（带宽管理）:限制每一个组群的带宽，让有限的带宽发挥最大的效用

## 6.拉流

- 直播协议选择 ：
  - 即时性要求较高或有互动需求的可以采用 RTMP , RTSP

- 对于有回放或跨平台需求的，推荐使用 `HLS`

• 直播协议对比：

直播协议	本质	原理	维护性	内容延迟	防火墙
RTMP	TCP长连接	每个时刻的数据，收到后立刻转发	跨平台差 (Flash Player 以外，需要自己移植)	1~3秒	很多防火墙会墙掉
HTTP-FLV	HTTP长连接	每个时刻的数据，收到后立刻转发	跨平台差 (Flash Player 以外，需要自己移植)	1~3秒	不会墙掉
HLS	HTTP短连接	集合一段时间，生成ts文件，更新m3u8	跨平台好 (Html5可以直接播放)	>10 秒	不会墙掉

直播协议对比.png

- `HLS` :由Apple公司定义的用于实时流传输的协议,HLS基于HTTP协议实现，传输内容包括两部分，一是M3U8描述文件，二是TS媒体文件。可实现流媒体的直播和点播，主要应用在iOS系统
  - `HLS`是 以点播的技术方式 来实现直播
  - `HLS`是 自适应码率流播，客户端会根据网络状况自动选择不同码率的视频流，条件允许的情况下使用高码率，网络繁忙的时候使用低码率，并且自动在二者间随意切换。这对移动设备网络状况不稳定的情况下保障流畅播放非常有帮助。
  - 实现方法是服务器端提供多码率视频流，并且在列表文件中注明，播放器根据播放进度和下载速度自动调整。
- `HLS`与`RTMP`对比 :HLS主要是延时比较大，RTMP主要优势在于延时低
  - `HLS`协议的小切片方式会生成大量的文件，存储或处理这些文件会造成大量资源浪费
  - 相比使用RTSP协议的好处在于，一旦切分完成，之后的分发过程完全不需要额外使用任何专门软件，普通的网络服务器即可，大大降低了CDN边缘服务器的配置要求，可以使用任何现成的CDN,而一般服务器很少支持RTSP。
- `HTTP-FLV` :基于HTTP协议流式的传输媒体内容。
  - 相对于RTMP，HTTP更简单和广为人知，内容延迟同样可以做到1~3秒，打开速度更快，因为HTTP本身没有复杂的状态交互。所以从延迟角度来看，HTTP-FLV要优于RTMP。

- **RTSP** :实时流传输协议,定义了一对多应用程序如何有效地通过IP网络传送多媒体数据.
- **RTP** :实时传输协议,RTP是建立在UDP协议上的,常与RTCP一起使用,其本身并没有提供按时发送机制或其它服务质量 (QoS) 保证,它依赖于低层服务去实现这一过程。
- **RTCP** :RTP的配套协议,主要功能是为RTP所提供的服务质量 (QoS) 提供反馈,收集相关媒体连接的统计信息,例如传输字节数,传输分组数,丢失分组数,单向和双向网络延迟等等。

## 7.解码

### \* 7.1 解封装 \*

- **demuxing (分离)** :从视频流、音频流,字幕流合成的文件(容器格式 (FLV, TS))中,分解出视频、音频或字幕,各自进行解码。

### \* 7.2 音频编码框架 \*

- **fdk\_aac** :音频编码解码框架,PCM音频数据和AAC音频数据互转

### \* 7.3 解码介绍 \*

- **硬解码** :用GPU来解码,减少CPU运算
  - 优点:播放流畅、低功耗,解码速度快,
  - \* 缺点:兼容不好
- **软解码** :用CPU来解码
  - 优点:兼容好
  - \* 缺点:加大CPU负担,耗电增加、没有硬解码流畅,解码速度相对慢

## 8.播放

- **ijkplayer** ([https://github.com/Bilibili/ijkplayer?utm\\_source=tuicool&utm\\_medium=referral](https://github.com/Bilibili/ijkplayer?utm_source=tuicool&utm_medium=referral)): 一个基于FFmpeg的开源Android/iOS视频播放器
  - API易于集成;
  - 编译配置可裁剪,方便控制安装包大小;
  - 支持硬件加速解码,更加省电
  - 简单易用,指定拉流URL,自动解码播放.

## 9.聊天互动

- IM:(InstantMessaging)即时通讯:是一个实时通信系统，允许两人或多人使用网络实时的传递文字消息、文件、语音与视频交流。
  - IM 在直播系统中的主要作用是实现观众与主播、观众与观众之间的文字互动。
- \* 第三方SDK \*

- 腾讯云 (<https://www.qcloud.com/doc/product/269>)：腾讯提供的即时通讯SDK，可作为直播的聊天室
- 融云 (<http://www.rongcloud.cn/>)：一个比较常用的即时通讯SDK，可作为直播的聊天室

## 五、如何快速的开发一个完整的iOS直播app

### 1、利用第三方直播SDK快速的开发

七牛云 ([http://www.qiniu.com/?utm\\_campaign=baiduSEM&utm\\_source=baiduSEM&utm\\_medium=baiduSEM&utm\\_content=baiduSEM](http://www.qiniu.com/?utm_campaign=baiduSEM&utm_source=baiduSEM&utm_medium=baiduSEM&utm_content=baiduSEM)):七牛直播云是专为直播平台打造的全球化直播流服务和一站式实现SDK端到端直播场景的企业级直播云服务平台。

\* 熊猫TV, 龙珠TV等直播平台都是用的七牛云

网易视频云 (<http://vcloud.163.com/live.html#bdpc>)：基于专业的跨平台视频编解码技术和大规模视频内容分发网络，提供稳定流畅、低延时、高并发的实时音视频服务，可将视频直播无缝对接到自身App。

### 2、第三方SDK公司为什么要提供SDK给我们？

- 希望把我们的产品和它绑在一条船上，更加的依赖它。
- 技术生钱，帮养一大批牛B的程序员

### 3、直播功能：自研还是使用第三方直播SDK开发？

第三方SDK开发：对于一个初创团队来讲，自研直播不管在技术门槛、CDN、带宽上都是有很大的门槛的，而且需要耗费大量的时间才能做出成品，不利于拉投资。

自研：公司直播平台大，从长远看，自研可以节省成本，技术成面比直接用SDK可控多了。

### 4.第三方SDK好处

- 降低成本
  - 使用好的第三方企业服务，将不用再花高价请猎头去挖昂贵的大牛，也不用去安抚大牛们个性化的脾气
- 提升效率
  - 第三方服务的专注与代码集成所带来的方便，所花费的时间可能仅仅是1-2个小时，节约近99%的时间，足够换取更多的时间去和竞争对手斗智斗勇，增加更大的成功可能性
- 降低风险
  - 借助专业的第三方服务，由于它的快速、专业、稳定等特点，能够极大地加强产品的竞争能力（优质服务、研发速度等），缩短试错时间，必将是创业中保命的手段之一
- 专业的事，找专业的人来做
  - 第三方服务最少是10-20人的团队专注地解决同一个问题，做同一件事情。第三方服务所带来的支持效果，绝不是通过1-2个人处理所能对比的，难道不是吗

➕ 推荐拓展阅读


📄 举报文章    © 著作权归作者所有


如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

¥ 打赏支持



♡ 喜欢 7


 分享到微博

 分享到微信

更多分享 ▼

41条评论 ( 按时间正序 · 按时间倒序 · 按喜欢排序 )

✎ 添加新评论

 R了个G (/users/d9431116937e)

2楼 · 2016-08-25 20:35 (/p/bd42bacbe4cc/comments/3837489#comment-3837489)

(/users/d9431116937e)

峥哥出品的都是精品！

♡ 喜欢(0)

回复



R了个G (/users/d9431116937e)

3 楼 · 2016-08-25 20:35 (/p/bd42bacbe4cc/comments/3837502#comment-3837502)

赞！

♡ 喜欢(0)

回复



Laki只是想做一个程序猿 (/users/4b72739c1379)

5 楼 · 2016-08-25 20:40 (/p/bd42bacbe4cc/comments/3837620#comment-3837620)

峥哥

♡ 喜欢(0)

回复



shannoon (/users/7f21e5ca7390)

6 楼 · 2016-08-25 20:41 (/p/bd42bacbe4cc/comments/3837647#comment-3837647)

牛逼牛逼

♡ 喜欢(0)

回复



iOS星客 (/users/6d4269a0560a)

7 楼 · 2016-08-25 20:43 (/p/bd42bacbe4cc/comments/3837689#comment-3837689)

牛逼

♡ 喜欢(0)

回复



Jesse\_Hello (/users/a9e59d4cbbd4)

8 楼 · 2016-08-25 20:44 (/p/bd42bacbe4cc/comments/3837698#comment-3837698)

666，加油，袁哥

♡ 喜欢(0)

回复



邈了个邈 (/users/ae32011d05fd)

11 楼 · 2016-08-25 20:55 (/p/bd42bacbe4cc/comments/3837977#comment-3837977)

niubi

♡ 喜欢(0)

回复





丁小时候 (/users/98f08a5f425c)

12 楼 · 2016-08-25 21:10 (/p/bd42bacbe4cc/comments/3838317#comment-3838317)

我说这么6,原来是 小码哥的 666

♡ 喜欢(0)

回复



奋斗的菜鸟 (/users/b5624824260c)

13 楼 · 2016-08-25 21:16 (/p/bd42bacbe4cc/comments/3838474#comment-3838474)

66666666

♡ 喜欢(0)

回复



MichaelHuyp (/users/2fba25c17a69)

14 楼 · 2016-08-25 21:36 (/p/bd42bacbe4cc/comments/3838990#comment-3838990)

支持👍

♡ 喜欢(0)

回复



yuzby (/users/115a6027f65e)

15 楼 · 2016-08-25 21:42 (/p/bd42bacbe4cc/comments/3839135#comment-3839135)

赞,太棒了,思维导图写的真好

♡ 喜欢(0)

回复



xiao公子 (/users/b79112ba0332)

16 楼 · 2016-08-25 23:05 (/p/bd42bacbe4cc/comments/3841181#comment-3841181)

哈哈😄,怀念小码哥的日子

♡ 喜欢(0)

回复



Calabash\_Boy (/users/f4dcd01eb5dd)

17 楼 · 2016-08-26 00:08 (/p/bd42bacbe4cc/comments/3842346#comment-3842346)

大赞!认真学习,期待后续!!

♡ 喜欢(0)

回复



I\_MT (/users/59df4965d888)

18 楼 · 2016-08-26 03:29 (/p/bd42bacbe4cc/comments/3843066#comment-3843066)

铮哥,你也说你是第一次接触,我想问问这么详细的资料是如何一步步搜集的,我觉得搜集资料才是一个人的本领!求指教。。。😄

♡ 喜欢(0)

回复

峥吖 (/users/b09c3959ab3b) : @I\_MT (/users/59df4965d888) 网上找资料 转化为自己的理解

2016.08.26 08:24 (/p/bd42bacbe4cc/comments/3844538#comment-3844538)

回复

✎ 添加新回复



CoderZb (/users/94deecdc8e45)

18 楼 · 2016.08.26 08:14 (/p/bd42bacbe4cc/comments/3844415#comment-3844415)

厉害呢

♡ 喜欢(0)

回复

加载更多 ▼ (/notes/5426165/comments?max\_id=3866064&order=asc&page=2)

写下你的评论...

发表



Ctrl+Enter 发表

被以下专题收入，发现更多相似内容：



**首页投稿 (/collection/bDHhpK)**

玩转简书的第一步，从这个专题开始。想上首页热门榜么？好内容想被更多人看到么？来投稿

(/collection/bDHhpK) 如果稿件也不要灰心哦~ 入选文章会进一个队...

93895篇文章 (/collection/bDHhpK) · 125436人关注



添加关注 (/collections/47/subscribe)



**iOS Developer (/collection/3233d1a249ca)**

分享 iOS 开发的知识，解决大家遇到的问题，讨论iOS开发的前沿，欢迎大家投稿

(/collection/3233d1a249ca) 11676篇文章 (/collection/3233d1a249ca) · 24894人关注



添加关注 (/collections/1276/subscribe)



**iOS学习 (/collection/1332c736fe39)**

学习从点滴开始！(PS: 拒绝部分投稿的文章仅仅是由于专题内已收录相关知识点的文章，并非是投稿的文章技术含量不够好，望谅解。)

(/collection/1332c736fe39)

4012篇文章 (/collection/1332c736fe39) · 4815人关注



添加关注 (/collections/19466/subscribe)