



HTML5视频直播那些事

@AlloyTeam 吕鸣

自我介绍



- 吕小鸣
- 项目：兴趣部落, now直播等
- 开源项目：MLogger, React Native下拉组件
- 技术擅长：web前端, ios开发
- 博客：<https://www.nihaoshijie.com.cn/>



大纲

- 移动直播的发展
- 视频基础知识
- H5与视频直播
- 直播性能优化
- 直播系统搭建

移动互联网的发展



移动端直播：下一个超级风口



更加实时的互动和交流

移动视频直播发展



- 移动端直播爆发
- H5承载着传播的重任

什么是视频?



视频基础知识

内容元素 (Content)

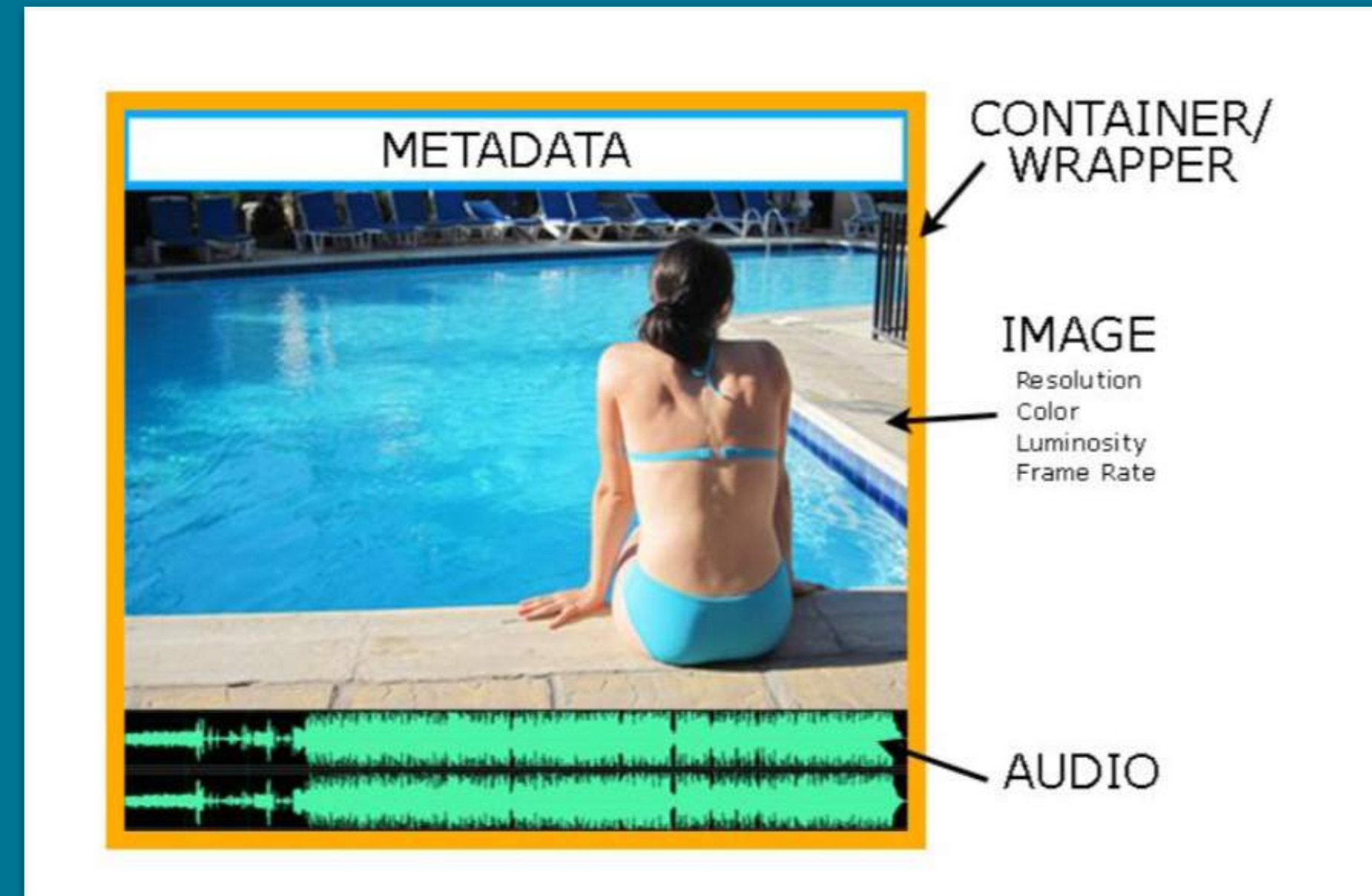
- 图像 (Image)
- 音频 (Audio)
- 元信息 (Metadata)

编解码器 (Codec)

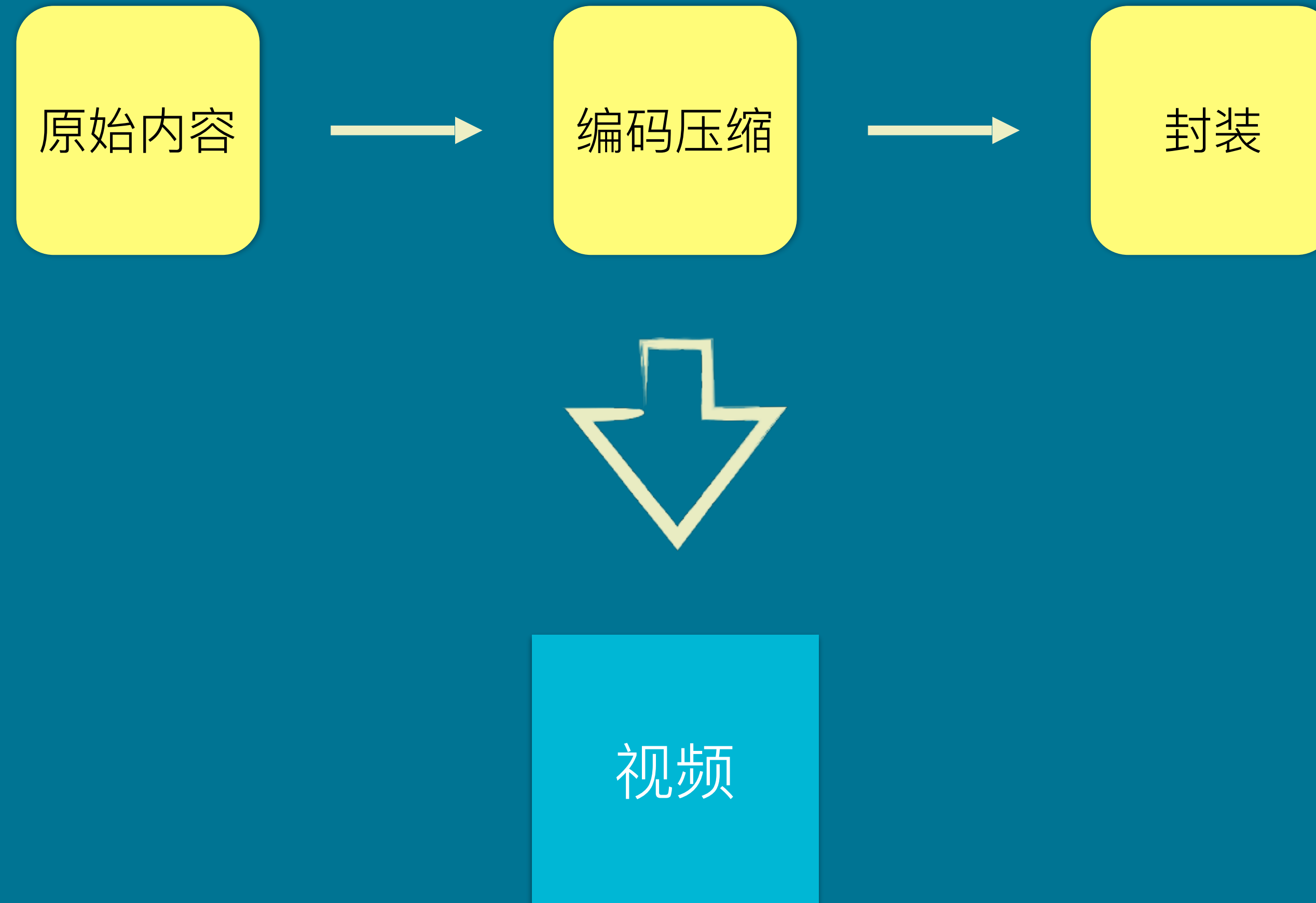
- 视频 (Video): H.263, H.264, H265...
- 音频 (Audio): AAC, HE-AAC...

容器文件格式 (Container)

- .MP3, MP4, FLV, AVI...



视频基础知识

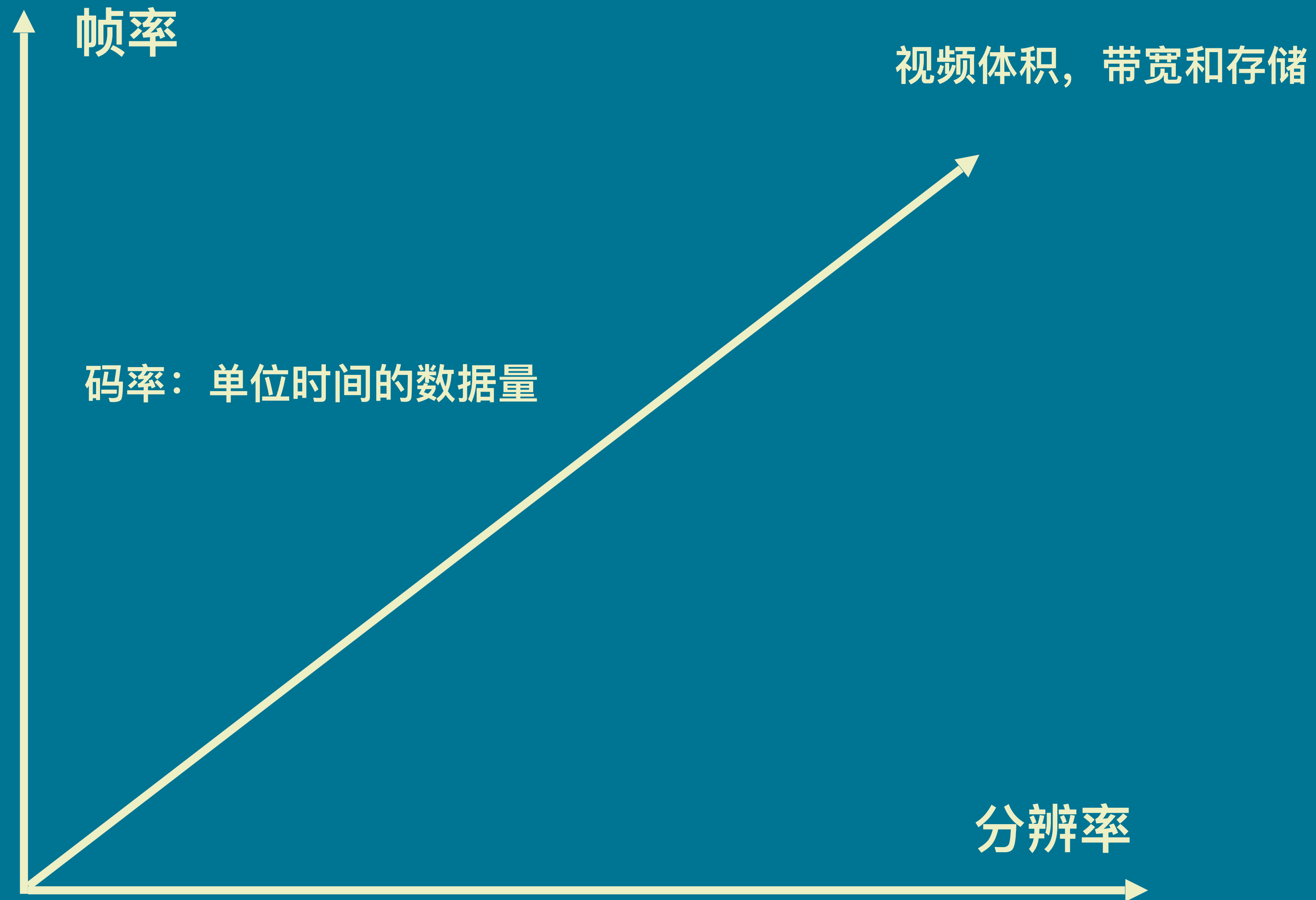


视频基础知识

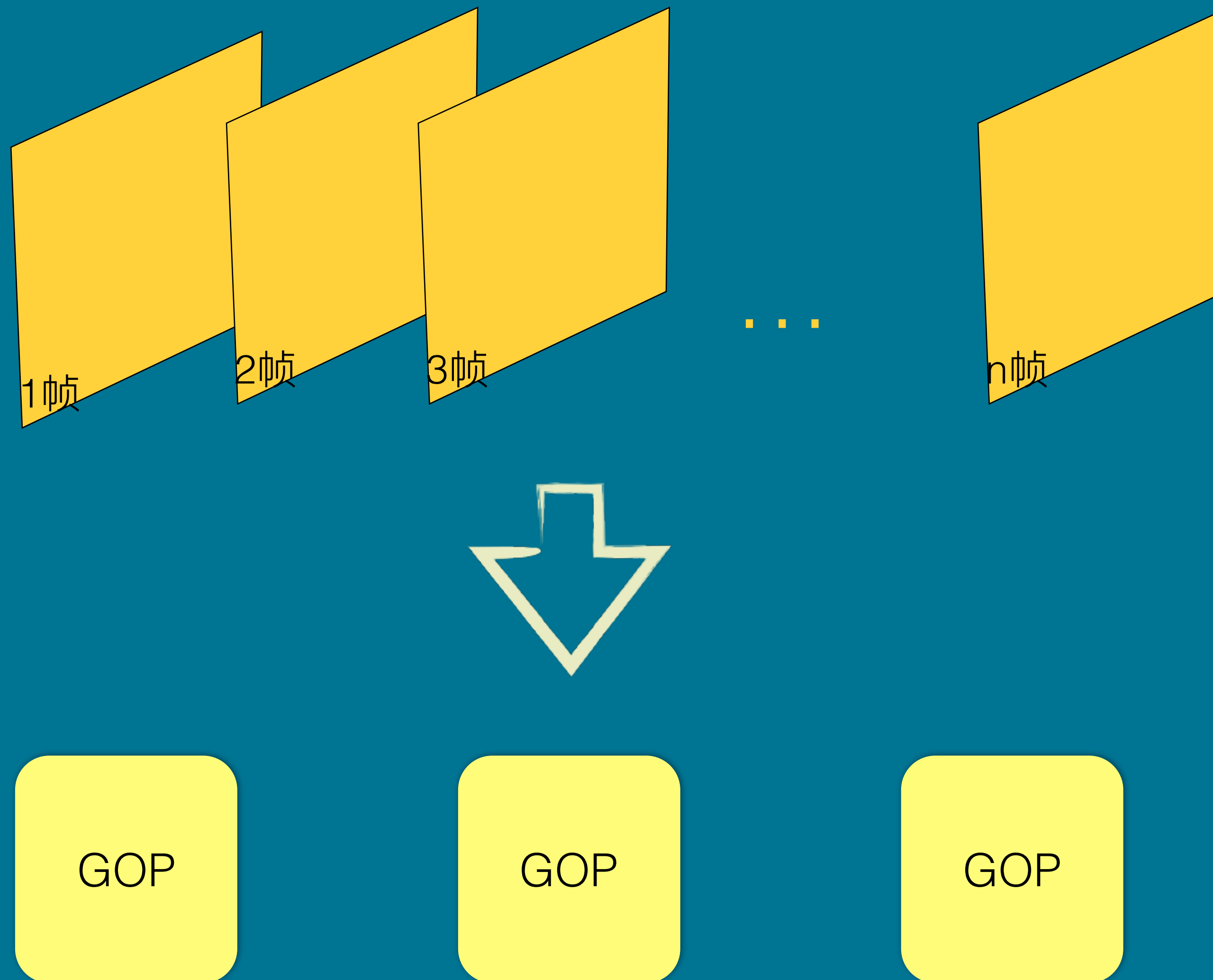
- 帧率 (Frame rate)
- 码率 (Bit rate)
- 分辨率 (Bit rate)
- 图片群组 (Group of Picture&GOP)



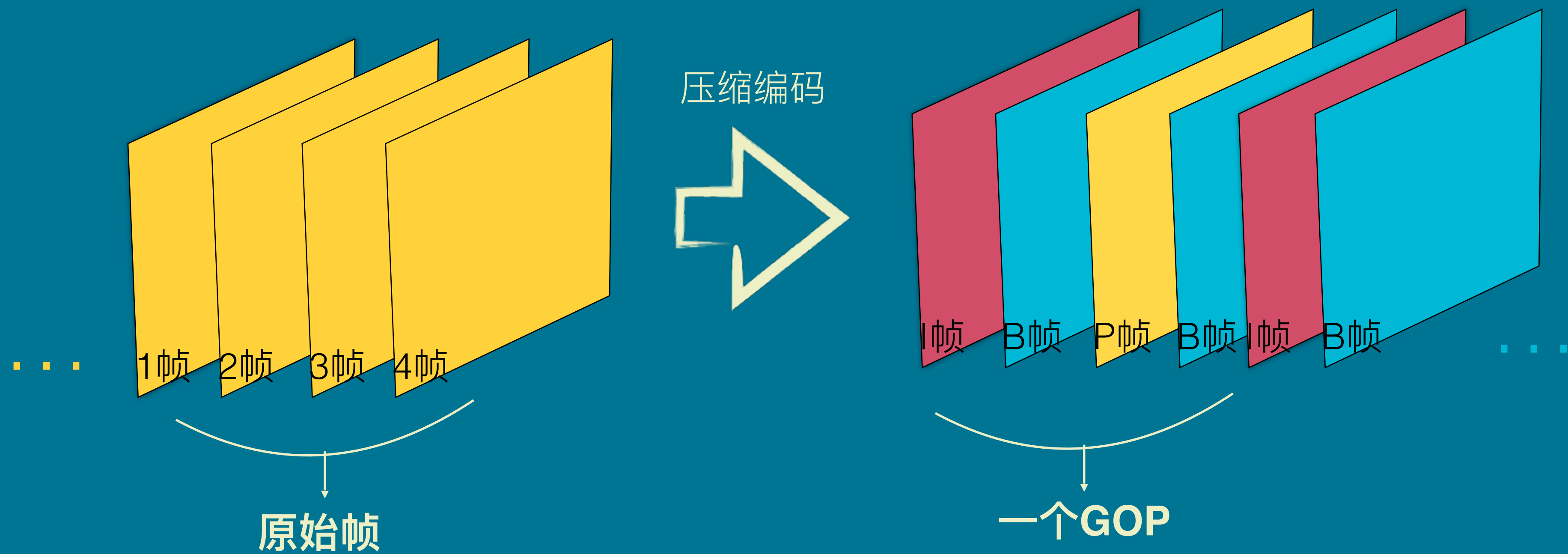
视频基础知识



视频基础知识



视频基础知识



GOP

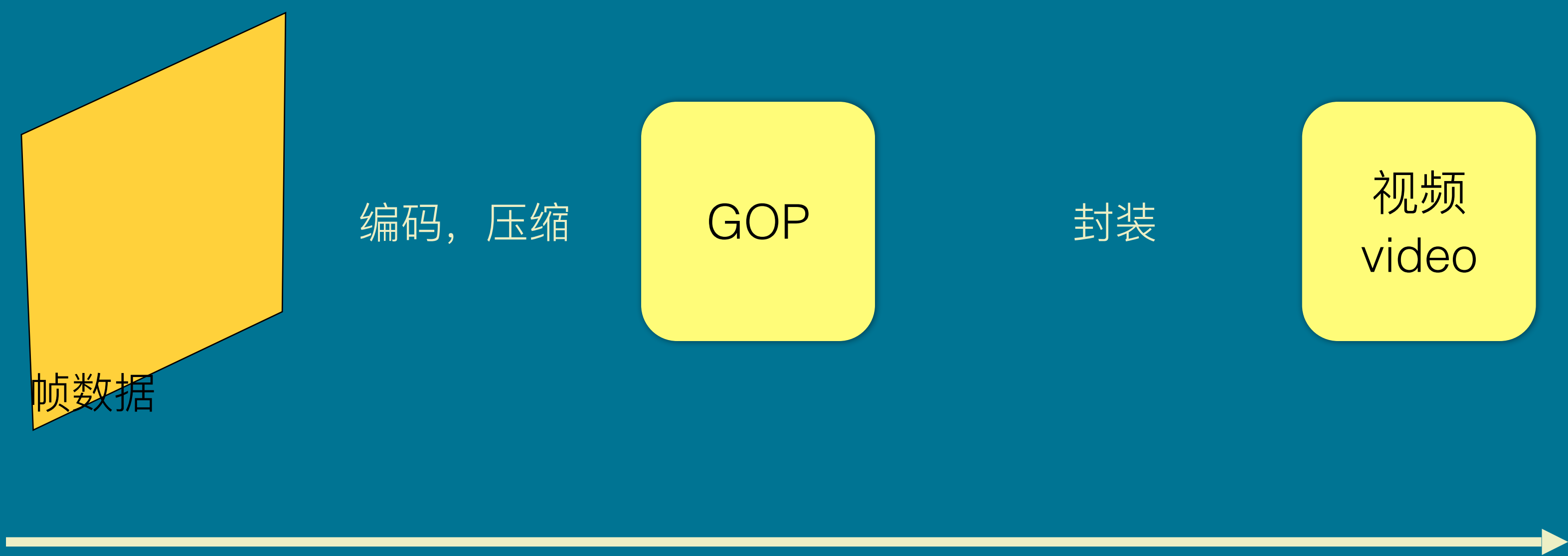
- **I帧**: 关键帧
- **B / P帧**: 参考帧

视频基础知识

GOP，关键帧，参考帧

帧类型	体积	解码依赖
I帧	大	无
P帧	小	前帧
B帧	小	前后帧

视频基础知识





那么，HTML5怎么播放视频呢？

H5播放视频



- 功能强大
- PC霸主

H5播放视频

Flash

- 移动端兼容性差
- 可以播放直播视频，延迟低
- 学习Flash开发语言，有一定成本。

```
<video>  
</video>
```

- 移动端兼容性好
- 可以播放直播视频，延迟高
- 提供基于js的api，前端开发更容易。

日常使用video标签时遇到过哪些问题呢？

video标签常见问题

视频内联播放

- ios的Safari中无法内联，webview可以定制

webview.allowsInlineMediaPlayback=YES

<video webkit-playsinline src=""/>

- android天然支持内联播放，但不排除部分机型使用系统播放器播放。

video标签常见问题

视频自动播放

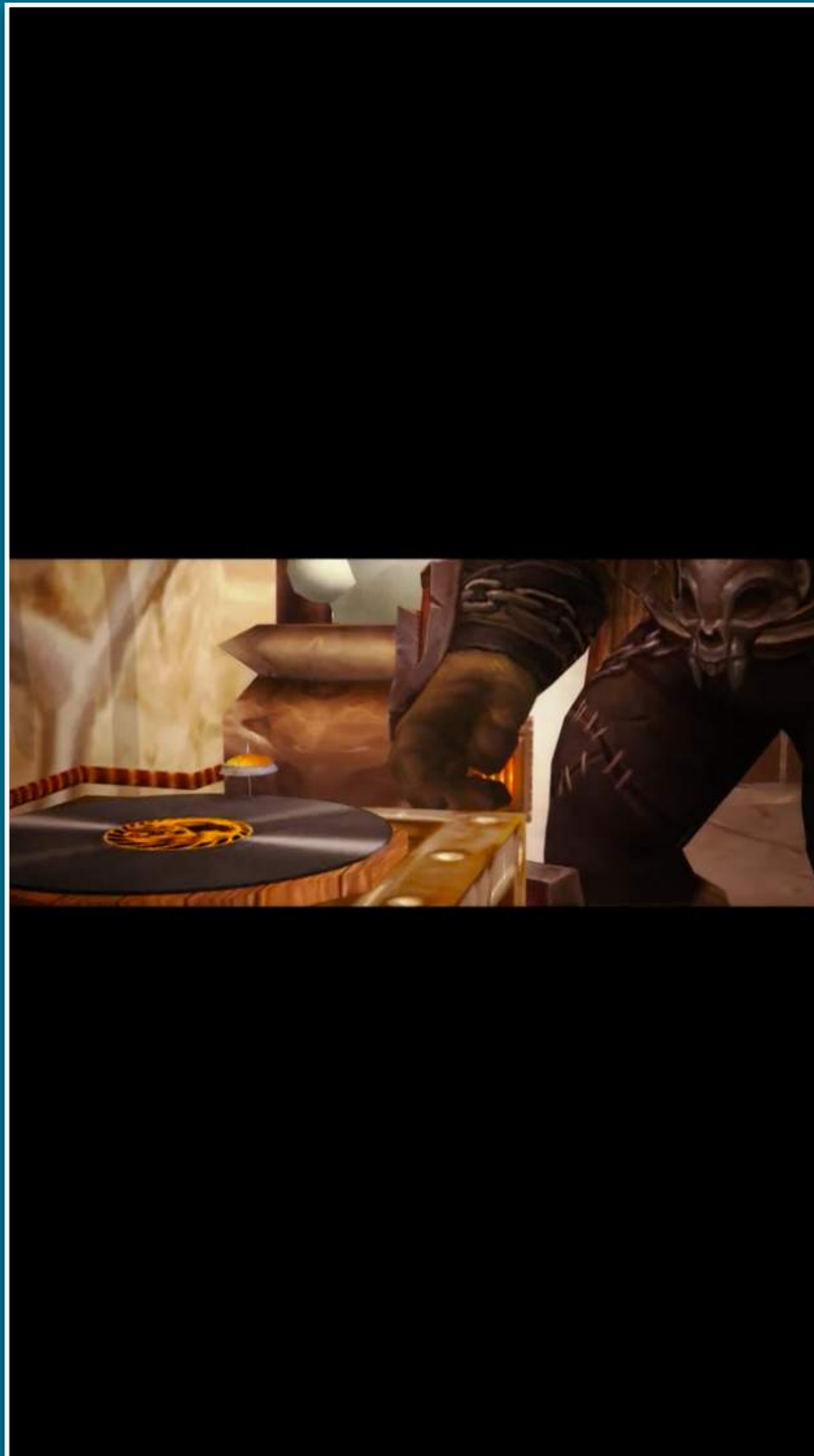
- 设置autoplay属性
- 事件回调里执行play()

webview.mediaPlaybackRequiresUserAction=NO

video标签常见问题

视频宽高设置

- 保持其纵横比



video标签常见问题

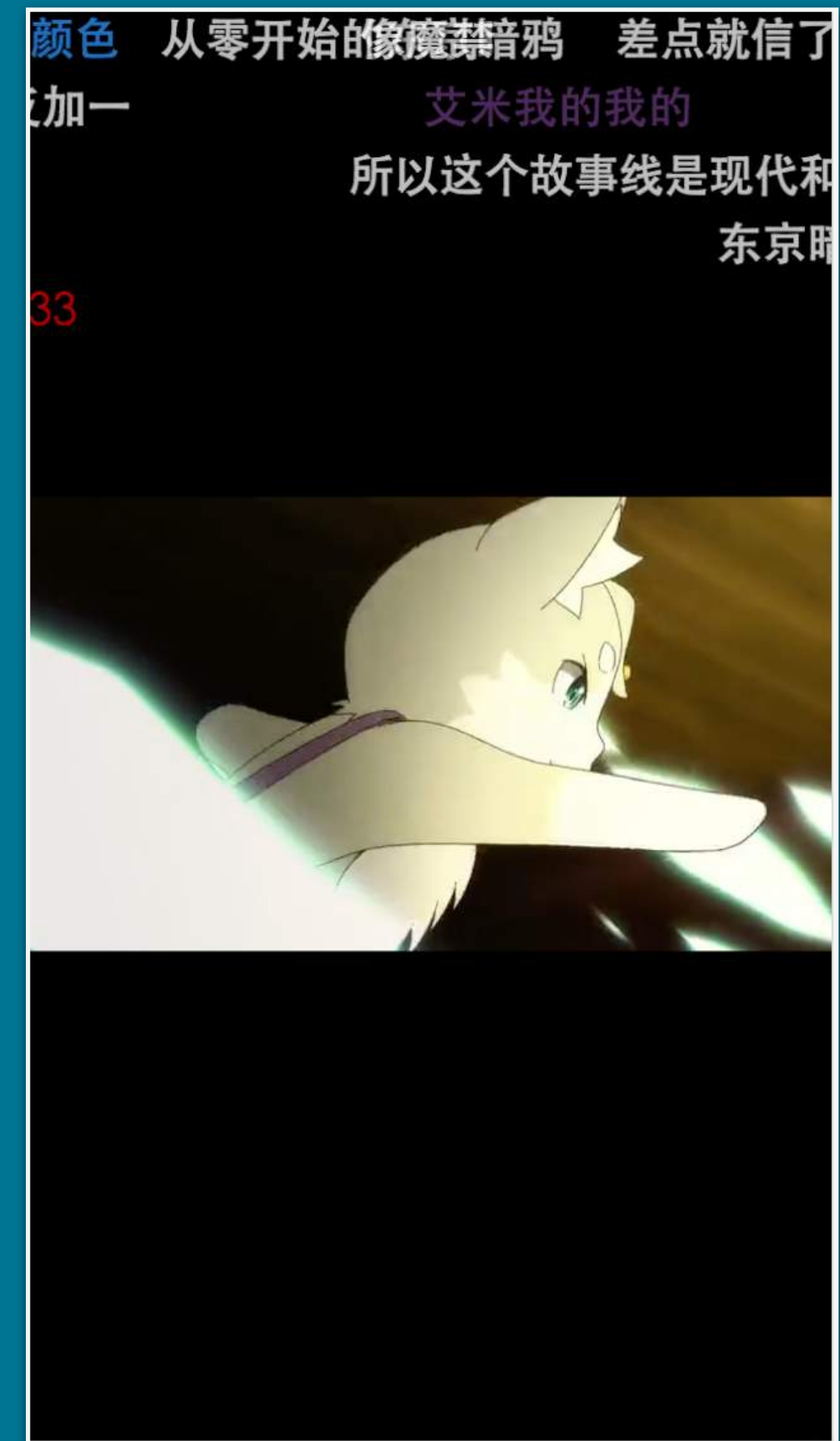
视频全屏问题

- **伪全屏：**通过设置样式和布局将视频内容撑满屏幕，无法覆盖webview。
- **真全屏：**一般是调用系统的native播放器来达到全屏观看，覆盖在webview之上。

video.webkitEnterFullScreen()

webkitbeginfullscreen事件

webkitendfullscreen事件



video标签常见问题

视频常用事件

事件	描述
loadstart	浏览器开始在网上寻找媒体数据
durationchange	播放时长被改变
loadedmetadata	浏览器获取完毕媒体的时间长和字节数
loadeddata	当当前帧的数据已加载，但没有足够的数据来播放指定音频/视频的下一帧时，会触发。
progress	当浏览器正在下载指定的视频时，会触发。
canplay	当浏览器能够开始播放指定的视频时，会触发。
canplaythrough	当浏览器预计能够在不停下来进行缓冲的情况下持续播放指定的视频时，会触发。
playing	当视频在已因缓冲而暂停或停止后已就绪时,会触发。
timeupdate	当目前的播放位置已更改时会触发。

video开源组件推荐



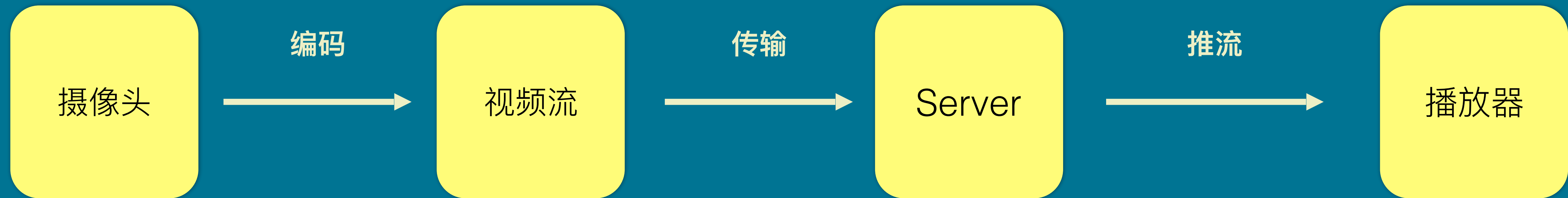
说了这么多基础知识，到底直播
是什么流程呢？

视频直播结构

- 动态改变的视频。
- 实时传输。
- 边生产，边消费。



视频直播结构



视频直播结构

- 视频录制端：

Native, webRTC

- 视频播放端：

flash, <video></video>, Native

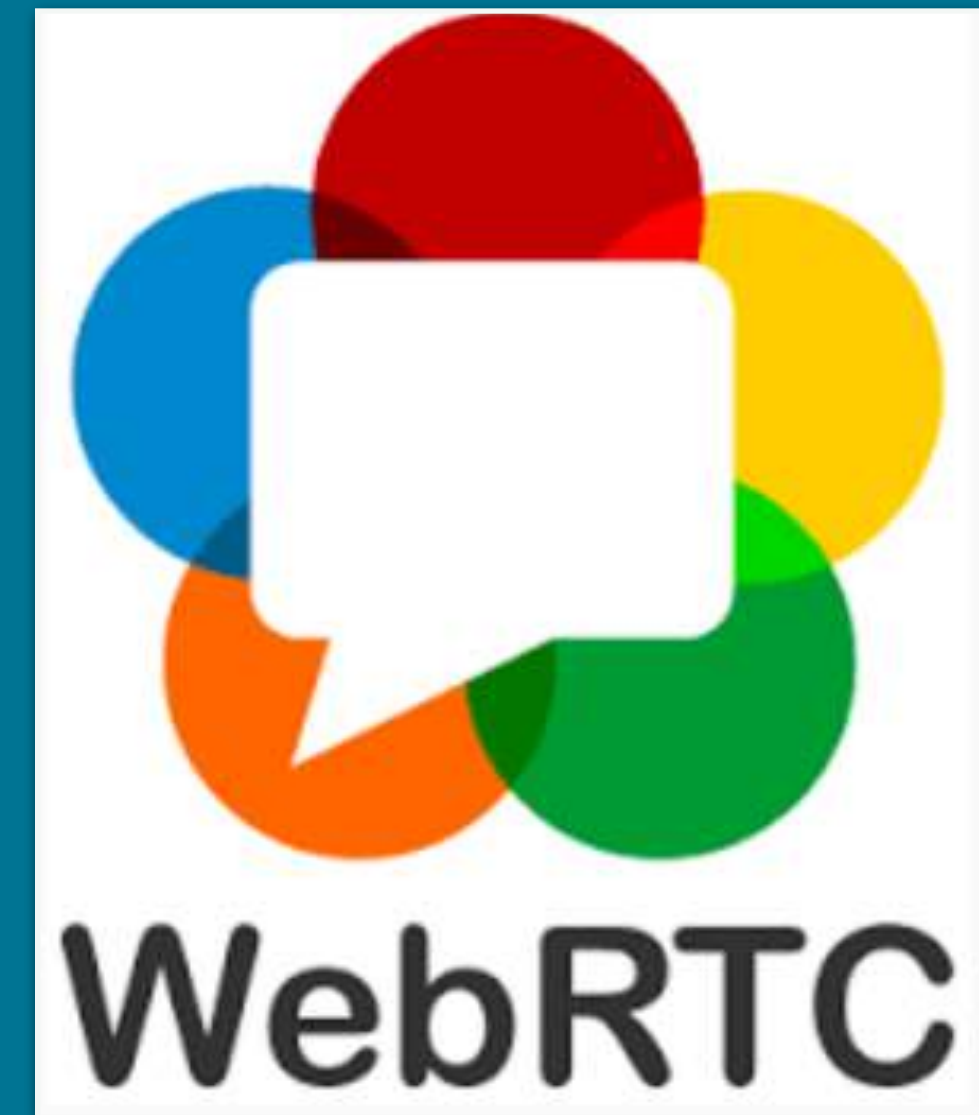
- 视频服务器端：

nginx

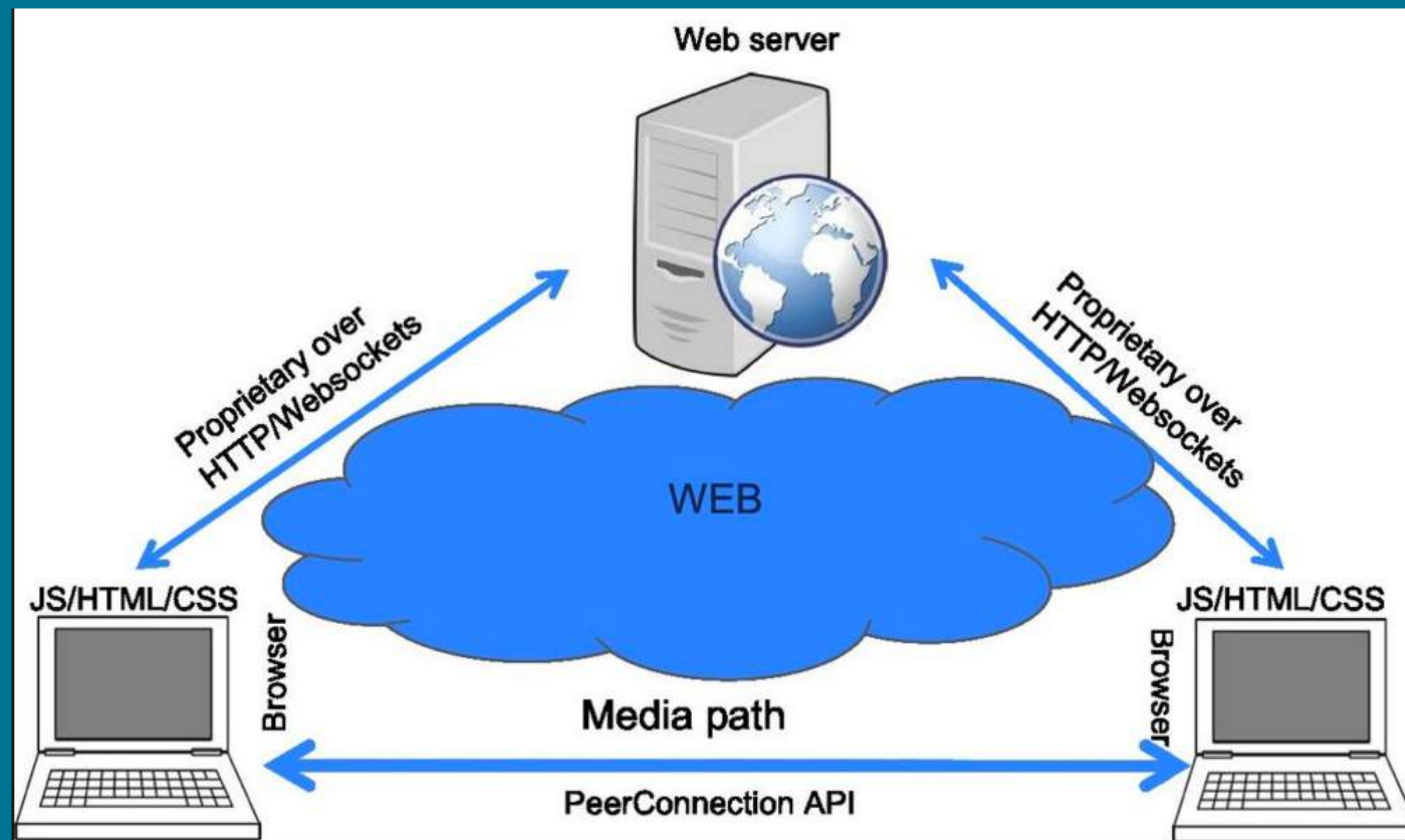
H5 录制视频

webRTC (Web Real-Time Communication)

- 提供js的api获取视频数据。
- 利用 webrtc 数据交互。



H5 录制视频



H5 录制视频

- webRTC兼容性

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51			9.2		4.4	
8	13	47	52			9.3		4.4.4	
11	14	48	53	9.1	39	10	all	52	51
		49	54	10	40				
		50	55	TP	41				
		51	56						

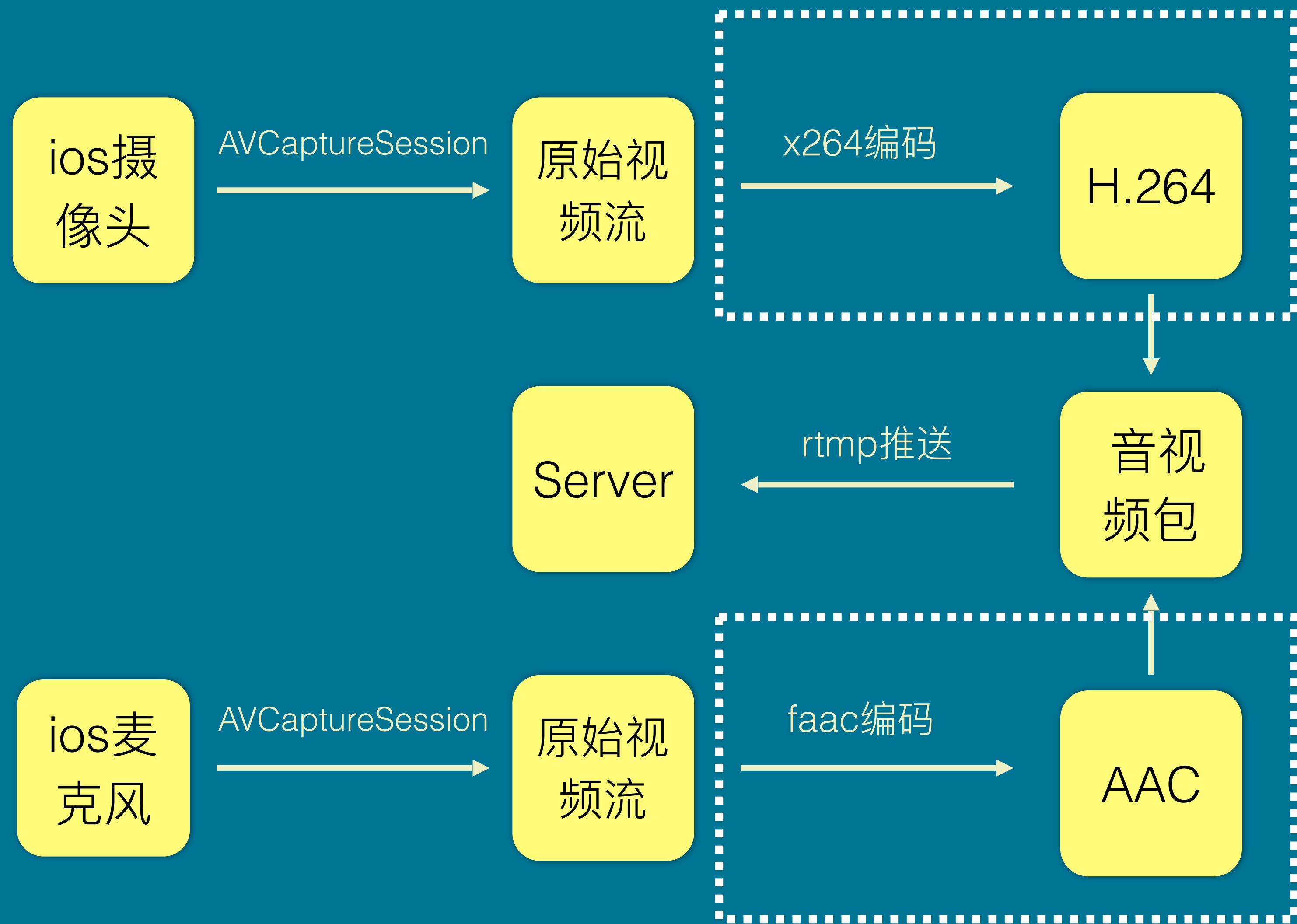
不适合做移动直播

Native 端录制视频

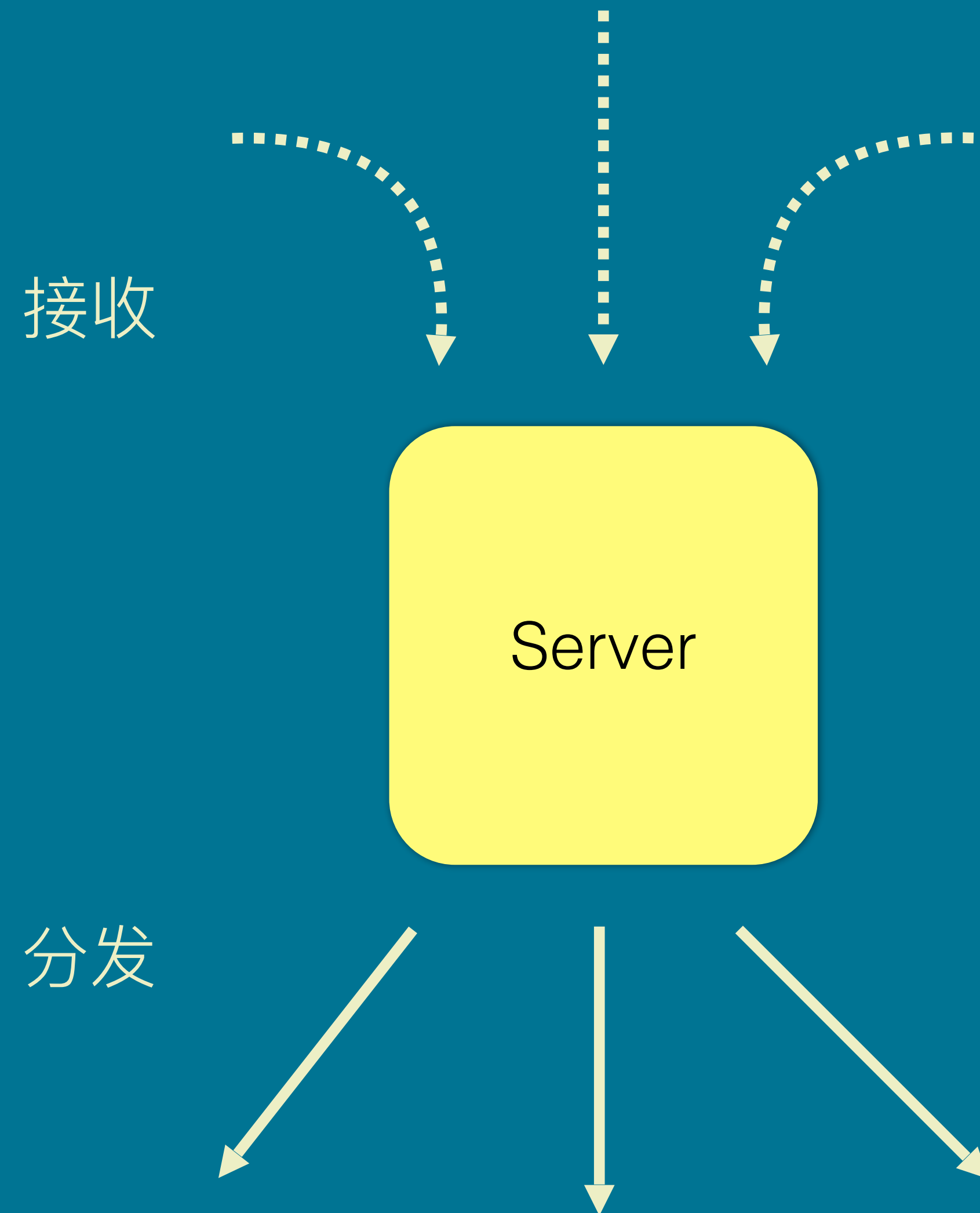
- 系统提供丰富的API接口可以调用。
- 提供丰富的编解码功能。
- 性能比web页面要好。



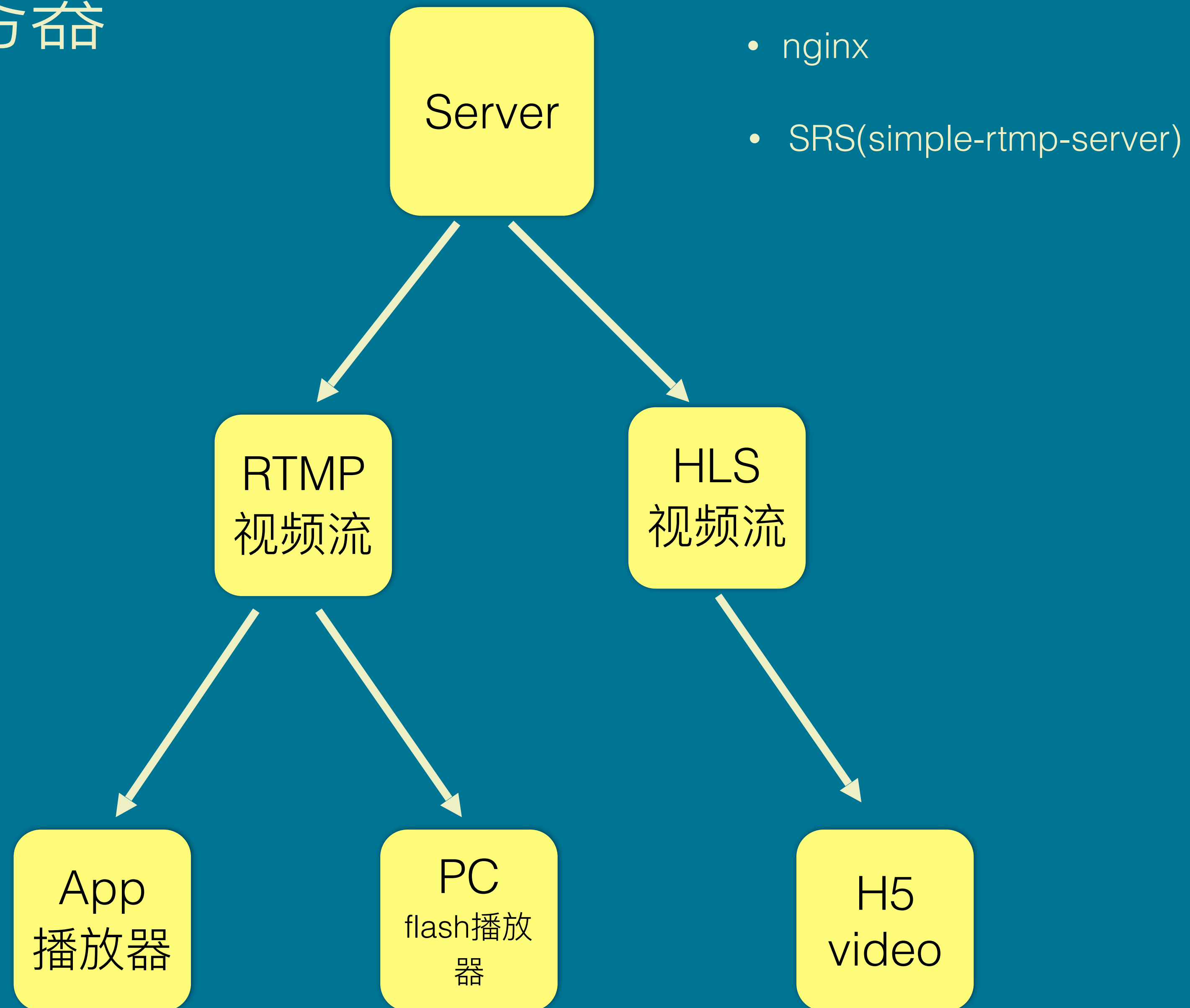
采集流程



推(拉)流服务器



推(拉)流服务器



播放直播视频

- HTML5使用<video/>



基于HLS协议

- PC端使用flash
- Native端使用系统播放器



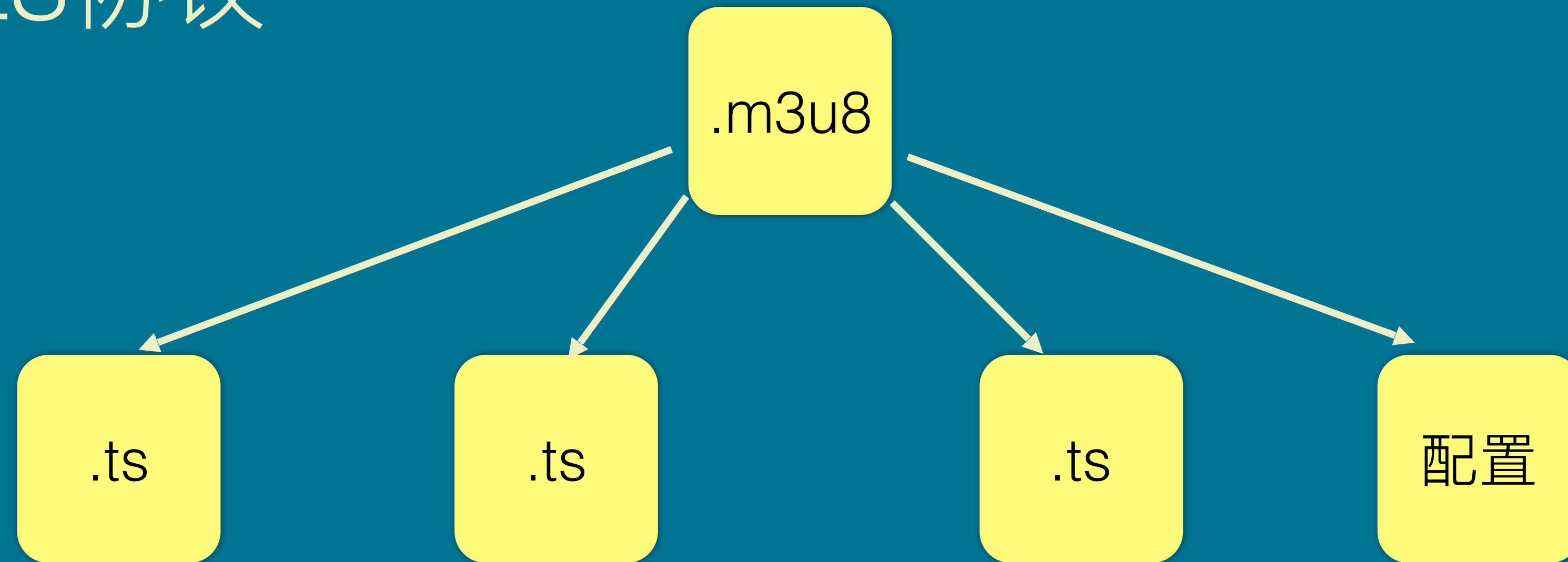
基于RTMP协议

H5 播放直播视频

- HLS(HTTP Live Streaming)协议播放直播流。
- android和ios天然支持

```
<video controls autoplay>  
  <source src="http://10.66.69.77:8080/hls/mystream.m3u8"  
type="application/vnd.apple.mpegurl" />  
  <p class="warning">Your browser does not support HTML5 video.</p>  
</video>
```

关于HLS协议



#EXTM3U

#EXT-X-MEDIA-SEQUENCE

#EXT-X-TARGETDURATION

#EXT-X-ALLOW-CACHE

#EXT-X-ENDLIST

#EXTINF

mystream-12.ts

m3u文件头

第一个TS分片的序列号

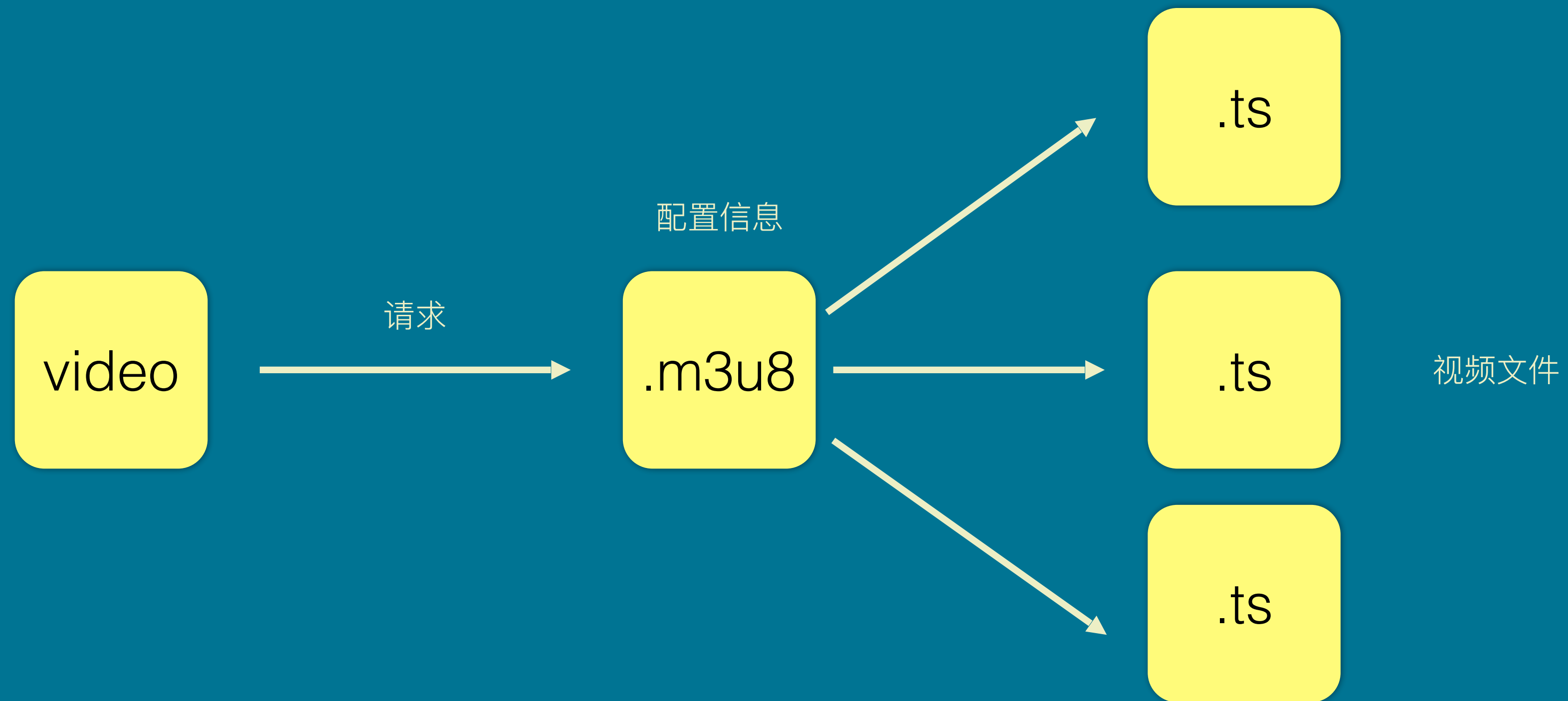
每个分片TS的最大的时长

是否允许cache

m3u8文件结束符

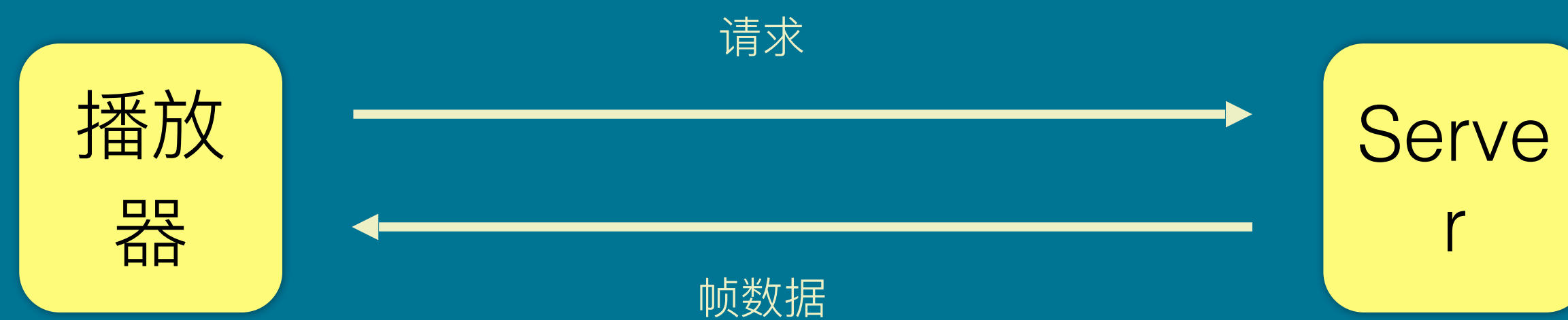
指定每个媒体段(ts)的持续时间（秒），仅对其后面的URI有效

HLS的请求流程



关于 RTMP

- **Real Time Messaging Protocol**（简称 RTMP）是 Macromedia 开发的一套视频直播协议，现在属于 Adobe。
- 和 HLS 一样都可以应用于视频直播，区别是 RTMP 基于 flash 无法在 iOS 的浏览器里播放，但是实时性比 HLS 要好。所以一般使用这种协议来上传视频流，也就是视频流推送到服务器。



关于 RTMP

	协议	原理	延时	优点	使用场景
RTMP	长链接 TCP	每个时刻收到数据后立刻发送	2s	延时低	即时互动
HLS	短链接 HTTP	集合一段时间数据生成ts分片文件 更新m3u8	10s－30s	跨平台	h5直播

H5直播实现



背景利用video播放视频

评论部分利用css和div渲染，同时利用websocket来实时获取评论并展示

点赞动画可以利用css3来实现

H5视频弹幕



H5视频弹幕

移动类型的弹幕要确定好位置，防止重叠。

移除视频外的弹幕要清理dom和重用

固定位置弹幕的出现和消失时间



整个video做为背景，弹幕层盖在上面

H5视频弹幕

- 弹幕文字使用translateX位移。
- 利用css3的transition-duration控制弹幕速度。
- 文字碰撞和重叠检测。
- websocket实时获取弹幕数据。

H5视频弹幕



移动直播性能指标

视频首屏打开耗时

直播视频的延迟

直播页面的交互性能

视频首屏打开耗时

- 页面打开耗时

1. 优化直播页业务逻辑。
2. 精简资源加载，减少和延迟加载非直播逻辑的资源。
3. 直播页直出。

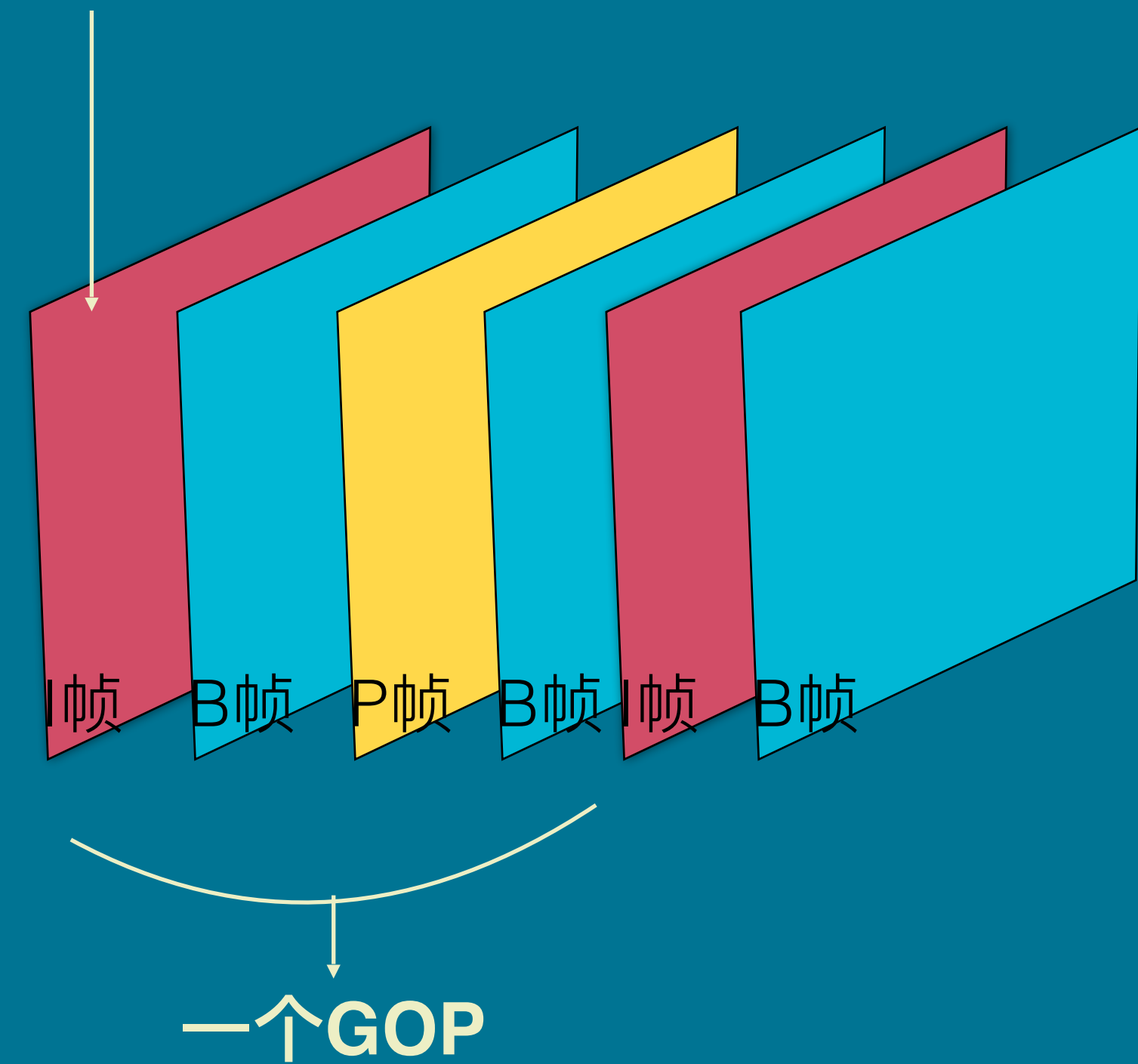
传统的H5页面优化策略同样适用于直播页面

直播视频秒开

- 修改GOP策略：

1. 播放视频逻辑优先发起，优先级最高。
2. 修改播放器逻辑，拿到关键帧即开始播放。
3. 在服务器端缓存上一个GOP。
4. 合理设置一个GOP时间。

修改播放器逻辑，拿到I帧则可以展示内容。



一般的播放器拿到一整个GOP才开始播放视频。

视频首屏打开耗时

- GOP的大小对视频的影响

	好处	坏处
GOP较大	体积小，更清晰	首屏慢，延迟高
GOP较小	首屏快，延迟低	体积大，更模糊

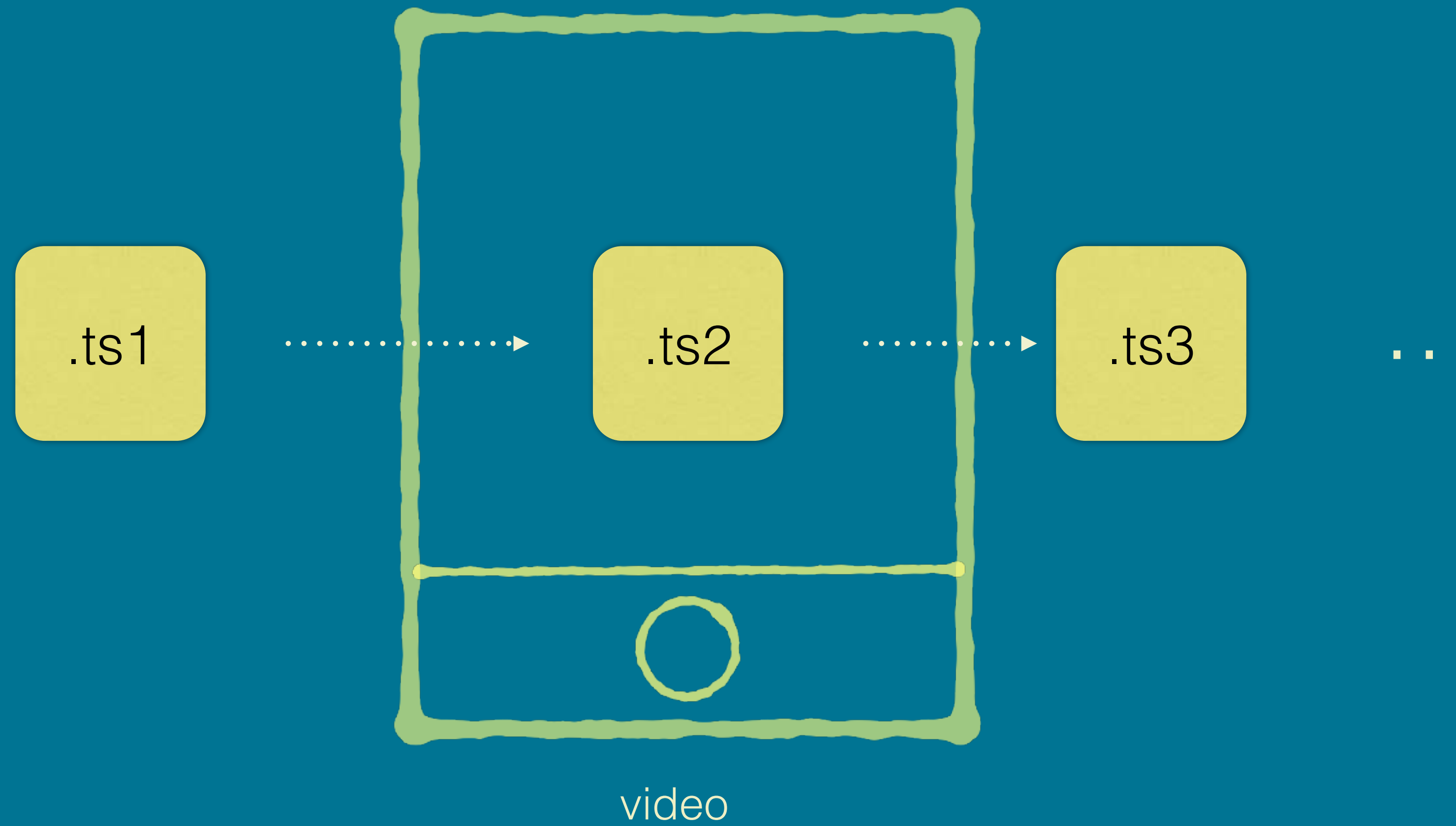
视频延迟

视频首屏打开耗时

直播视频的延迟

直播页面的交互性能

HLS直播延时



HLS直播延时

ts文件依次下载播放

ts切片时长大小设置

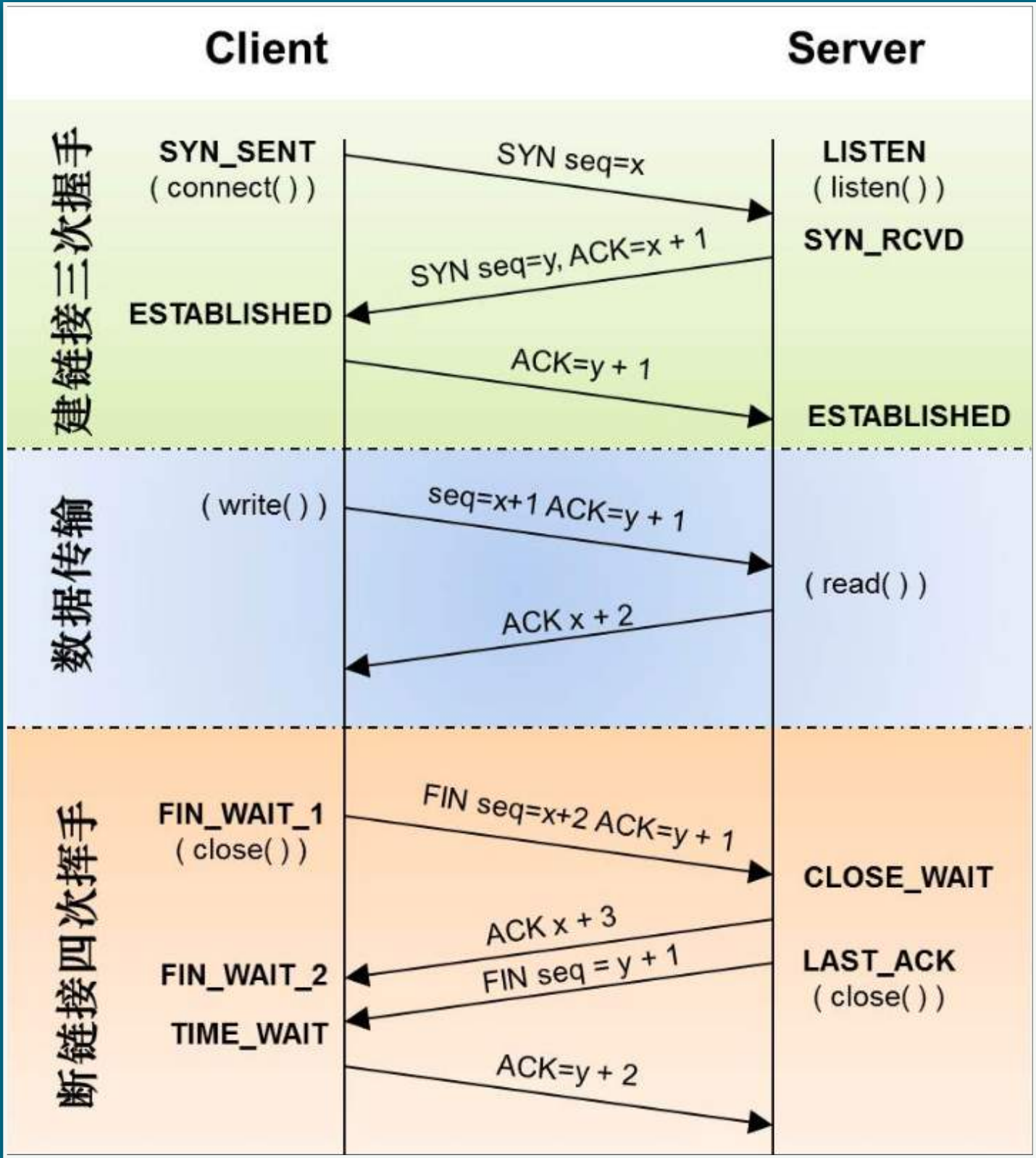
ts: 5s

延迟: 15s到20s

基于短链接的HTTP请求下载

HLS直播延时

HTTP请求的消耗：



- 3次握手，4次挥手
- 短链接，无状态

HLS直播延时

优化HTTP请求？

HTTP/2

For Faster and Safer Internet

- 二进制分帧
- 多路复用
- 头部压缩
- 服务端推送

HLS直播延时

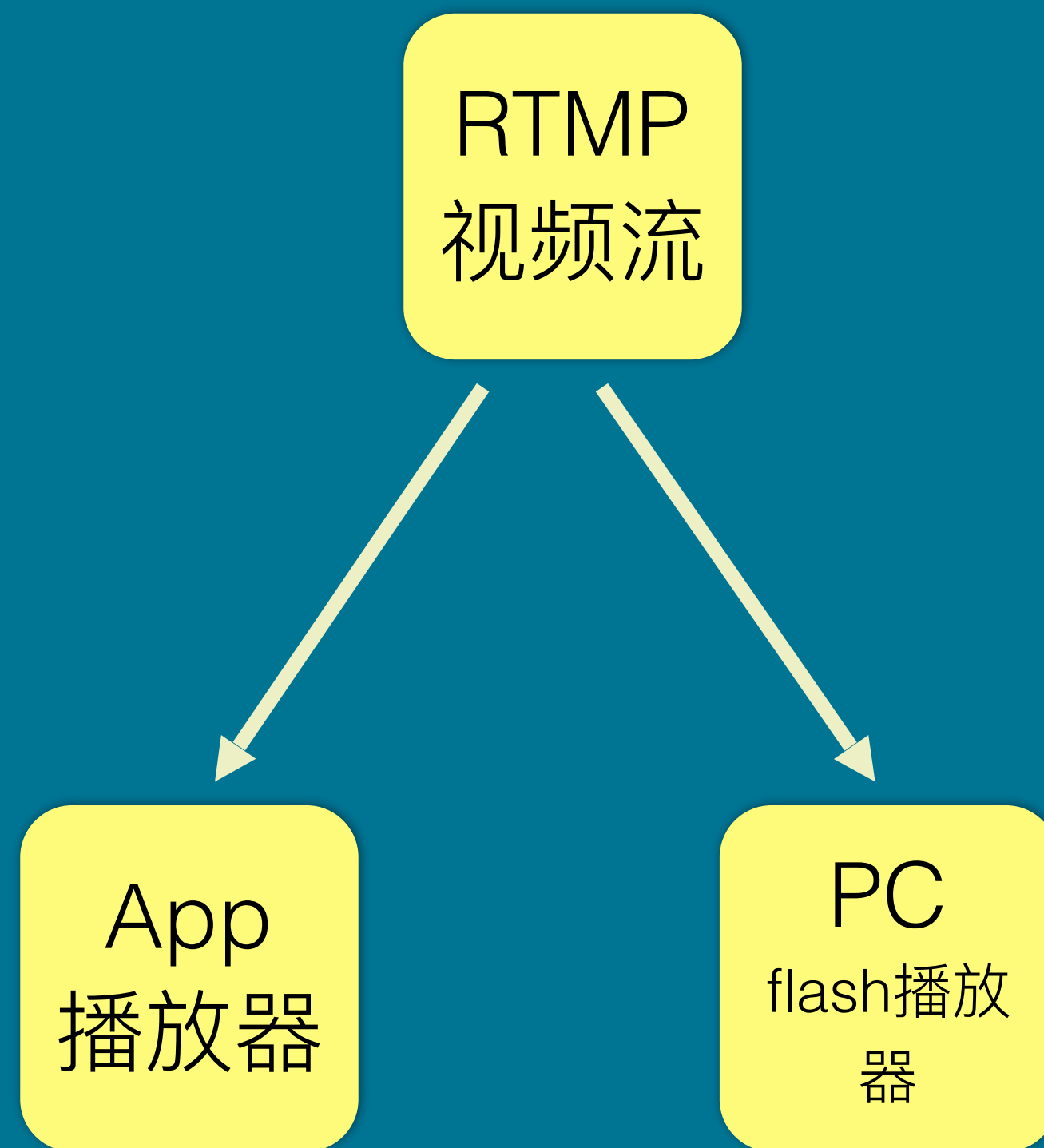
- 采用HTTP2.0

- 1. 适当减少ts文件长度来降低延迟。
- 2. 多路复用和头部压缩可以提升拉取ts文件的速度。

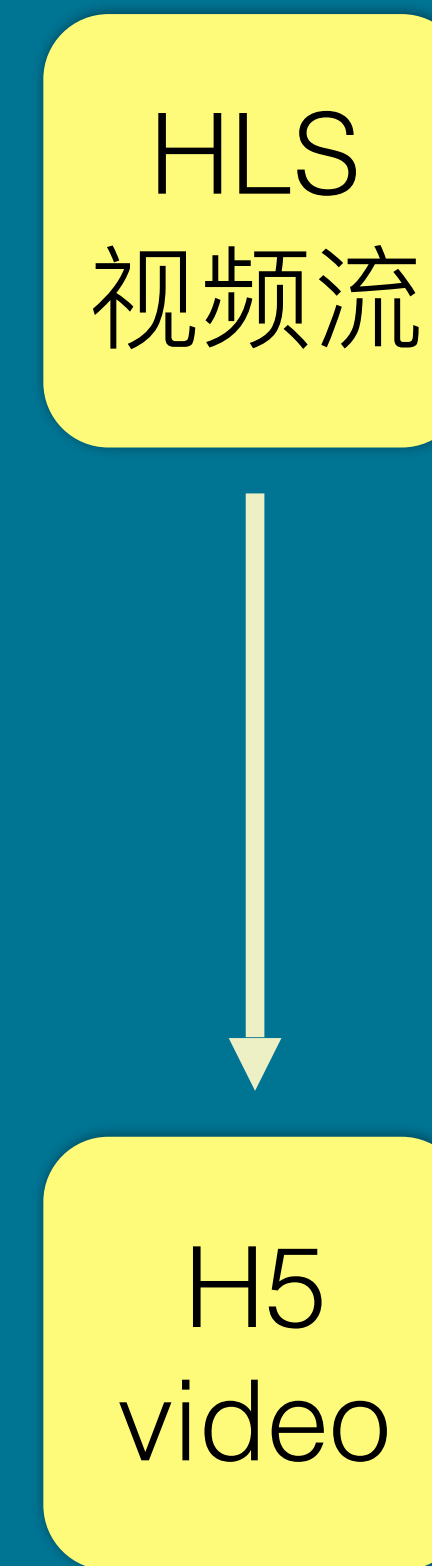
	相同清晰度
http2	延迟较低，更流畅
http1	延迟较高，更多缓冲

直播视频的延迟

- 选择合适的直播协议。



- 低延迟，稳定。



- 高延迟，益传播。

直播页交互性能

视频首屏打开耗时

直播视频的延迟

直播页面的交互性能

直播页面的交互性能。



• 评论内容数据获取:

- 使用webscoket。
- 使用长轮训。
- 定时器发起请求。

• 评论和点赞的dom渲染:

- dom及时清理。
- 使用局部滚动或者iscroll
- css3动画和canvas

直播页面的交互性能。

- 做好异常处理和错误提示。
- 直播的交互逻辑和视频的播放逻辑区分开来。
- 组件hybrid化。

搭建自己的直播服务

iOS App

推流App

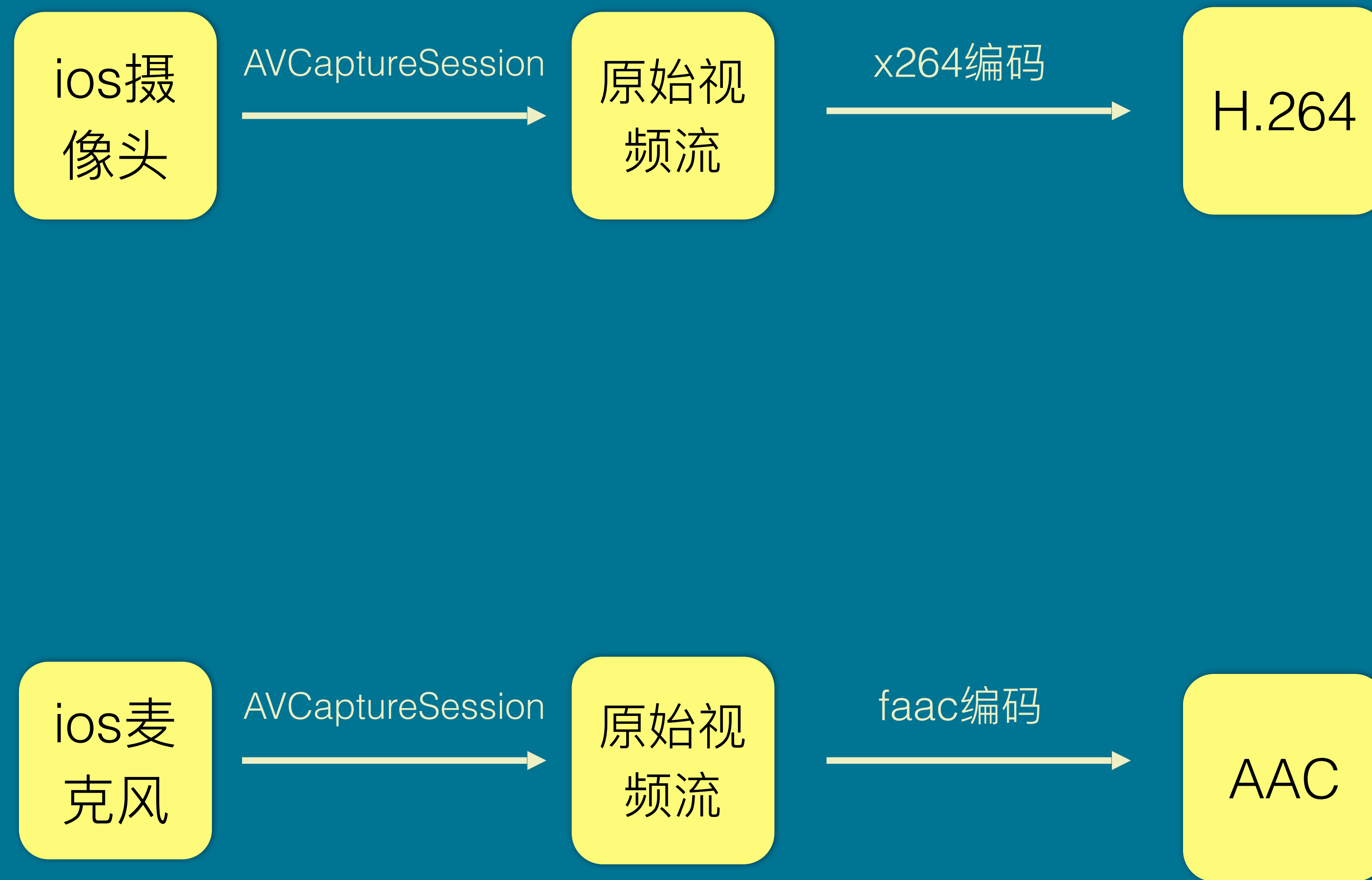
Nginx

推流服务器

HTML5

视频播放器

推流App



推流服务器

- 安装一台 nginx 服务器。
- 安装 nginx 的 RTMP 扩展，目前使用比较多的是 <https://github.com/arut/nginx-rtmp-module>
- 配置 nginx 的 conf 文件

```
rtmp {  
    server {  
        listen 1935; #监听的端口  
  
        chunk_size 4000;  
  
        application hls { #rtmp推流请求路径  
            live on;  
            hls on;  
            hls_path /usr/local/var/www/hls;  
            hls_fragment 5s;  
        }  
    }  
}
```

H5 播放直播视频

- 一个简单的HTML页面
- video标签设置直播流地址

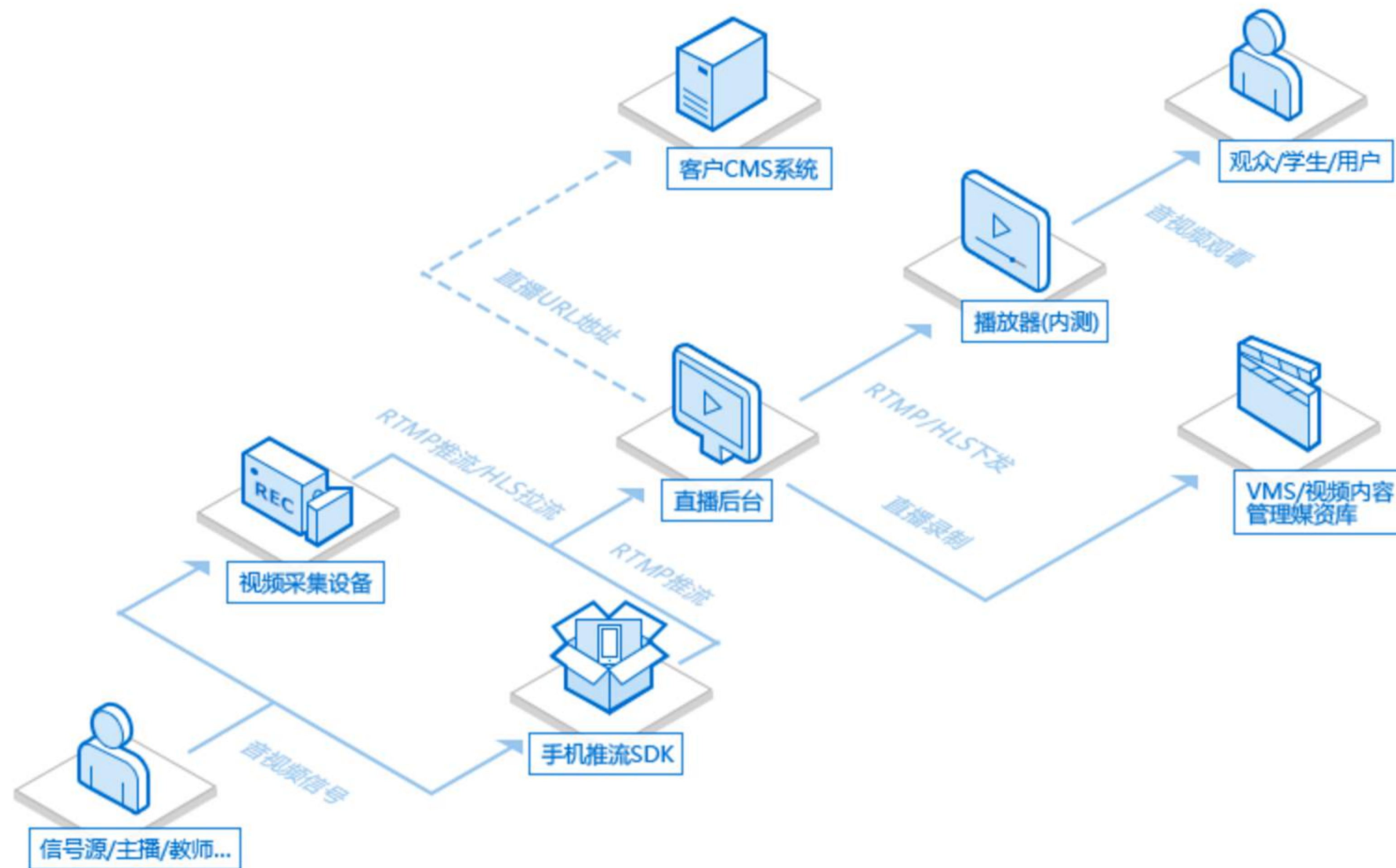
```
<video controls autoplay>  
  <source src="http://10.66.69.77:8080/hls/mystream.m3u8"  
type="application/vnd.apple.mpegurl" />  
  <p class="warning">Your browser does not support HTML5 video.</p>  
</video>
```

搭建自己的直播服务

- github地址：



选择第三方直播服务



Q&A