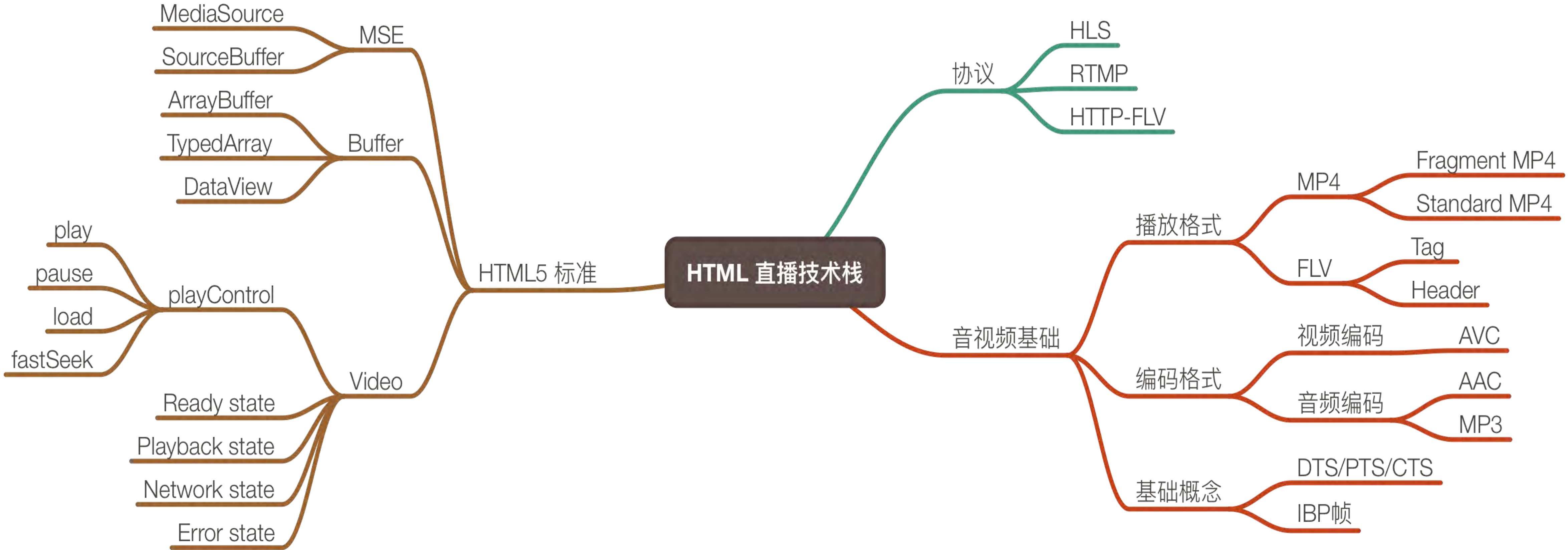


HTML5 直播技术栈





MSE 标准格式

MIME	支持格式	文件名
audio/webm video/webm	WebM Byte Stream Forma	.webm
audio/mp4 video/mp4	ISO BMFF Byte Stream Format	.mp4
audio/mp2t video/mp2t	MPEG-2 Transport Streams Byte Stream Format	.ts



● RTMP 协议



● HLS 协议

🎯 直播协议秒开



RTMP 协议解析

→ 使用方式:



```
<video  
  src="blob:http://egame.qq.com/9f837e14e28-9a08"  
</video>
```



```
<embed wmode="opaque"  
  src="rtmp://imgcache.qq.com/live/6721_d13dxfl1dddzd1233"  
  type="application/x-shockwave-flash"  
  pluginspage="http://get.adobe.com/cn/flashplayer/">
```

♥ RTMP 协议解析

→ RTMP 架构



♥ RTMP 协议解析

→ RTMP 架构



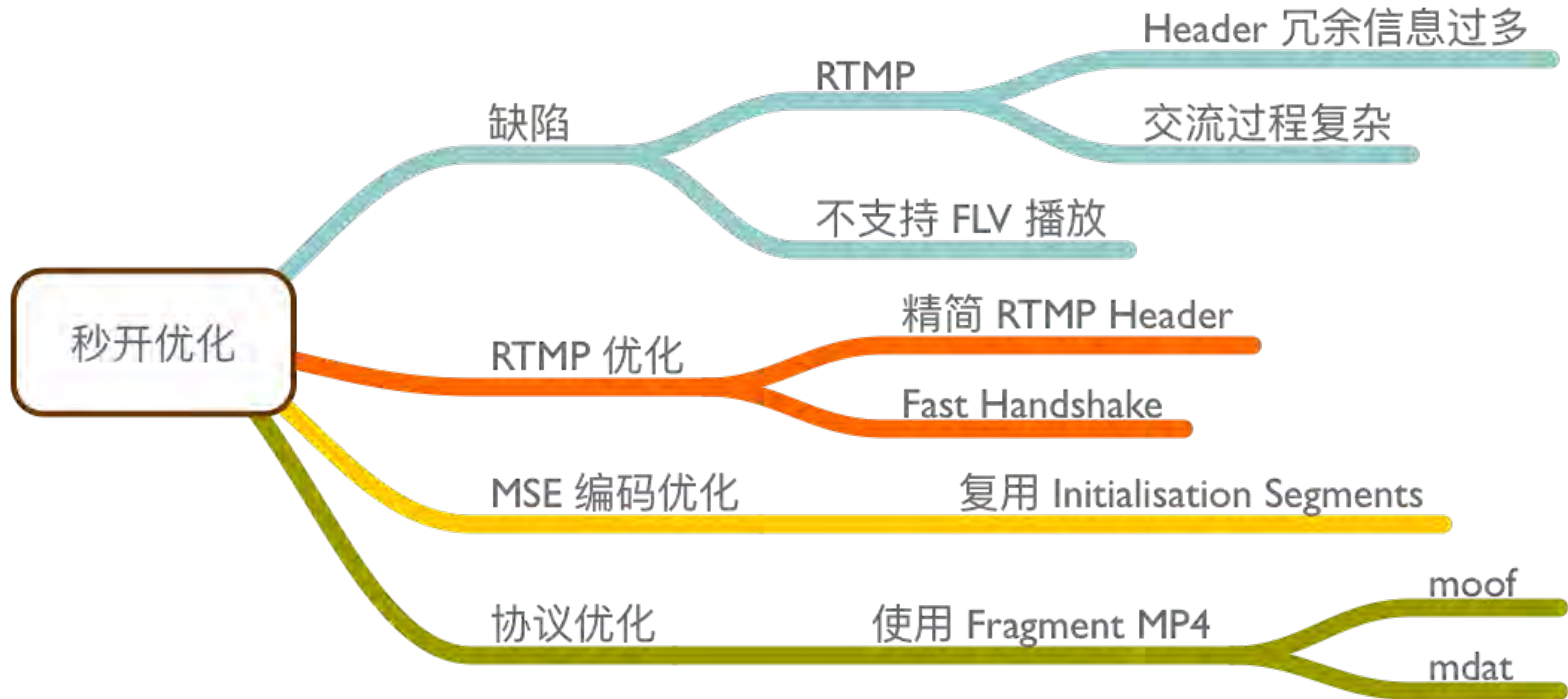
♥ RTMP 协议解析

→ RTMP 架构



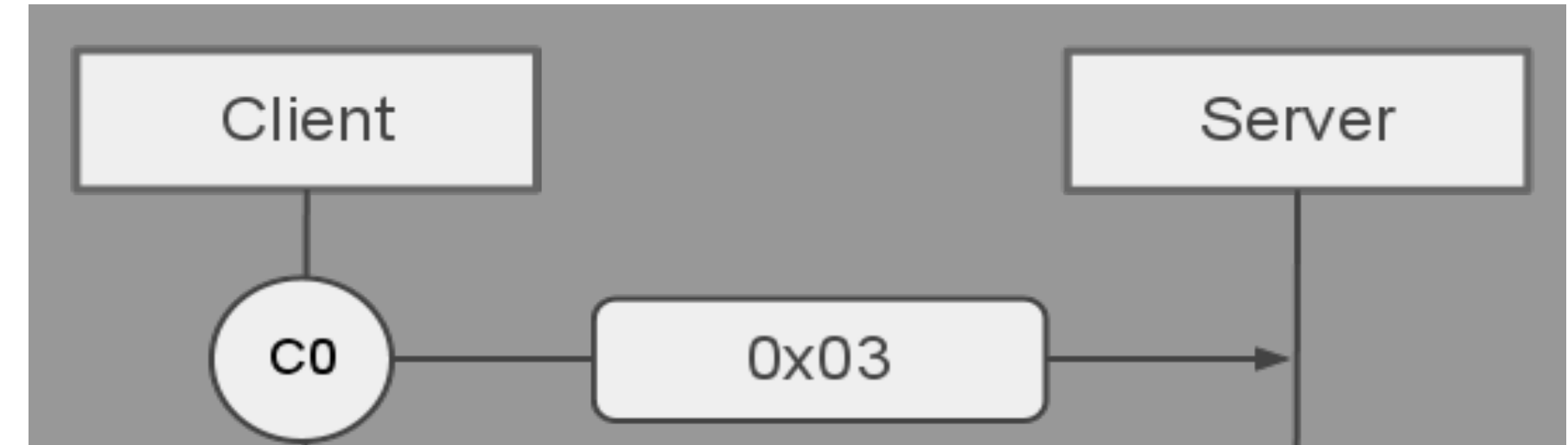
♥ RTMP 协议解析

→ 秒开优化

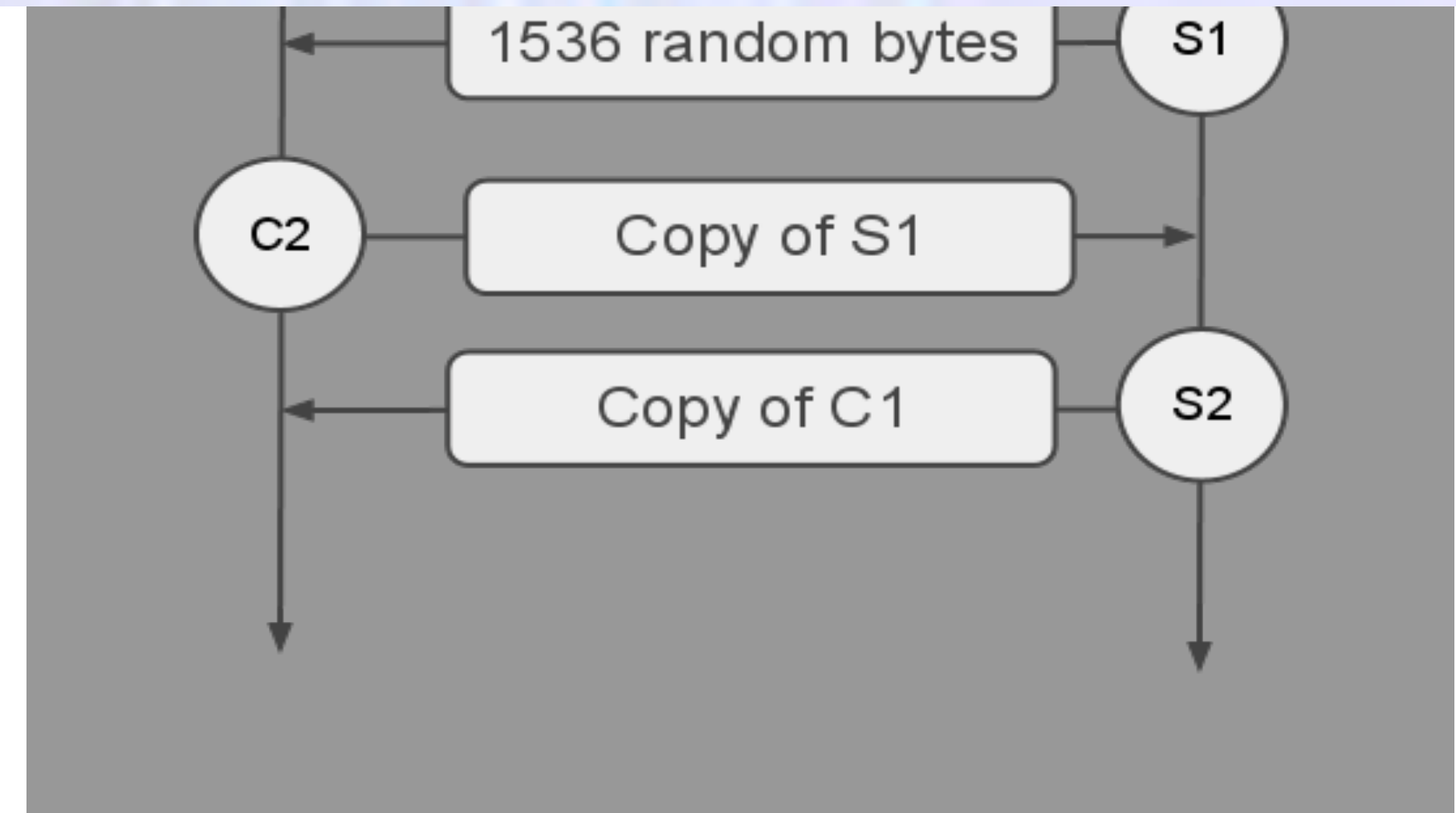


♥ RTMP 协议解析

→ 秒开优化

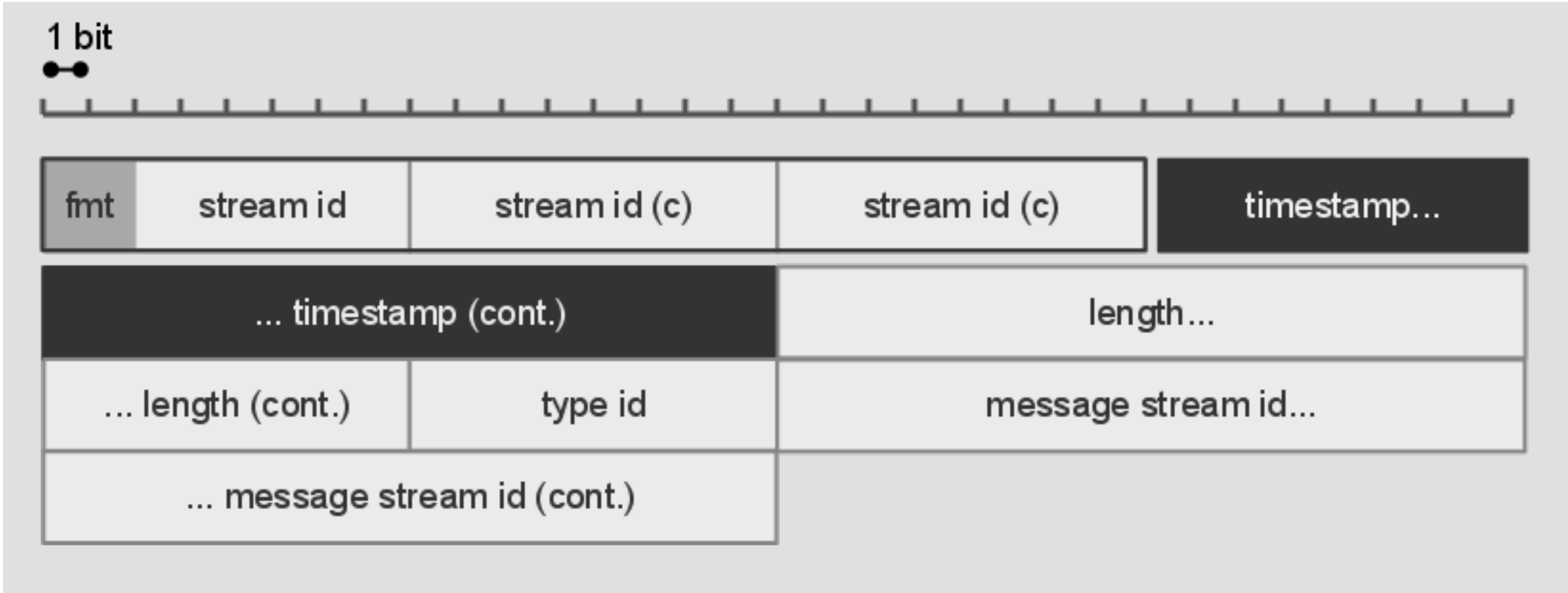


Handshake C0+C1
Handshake S0+S1+S2
Handshake C2



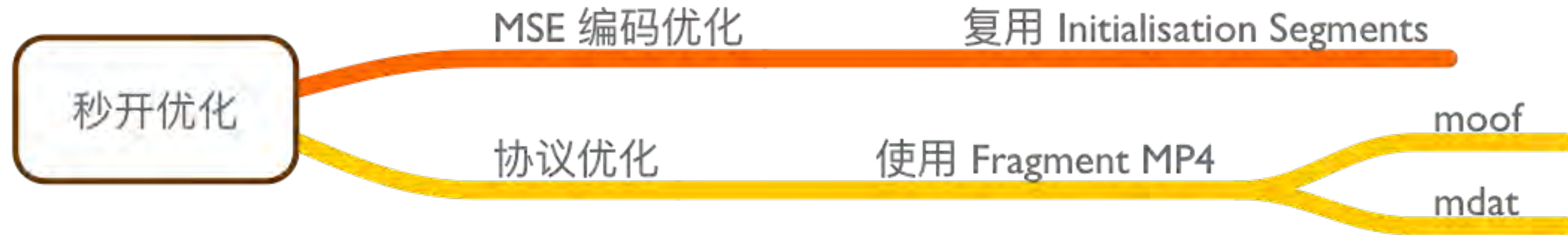
♥ RTMP 协议解析

→ 秒开优化



♥ RTMP 协议解析

→ 秒开优化



moof + mdat = Media Segments
Media Segments



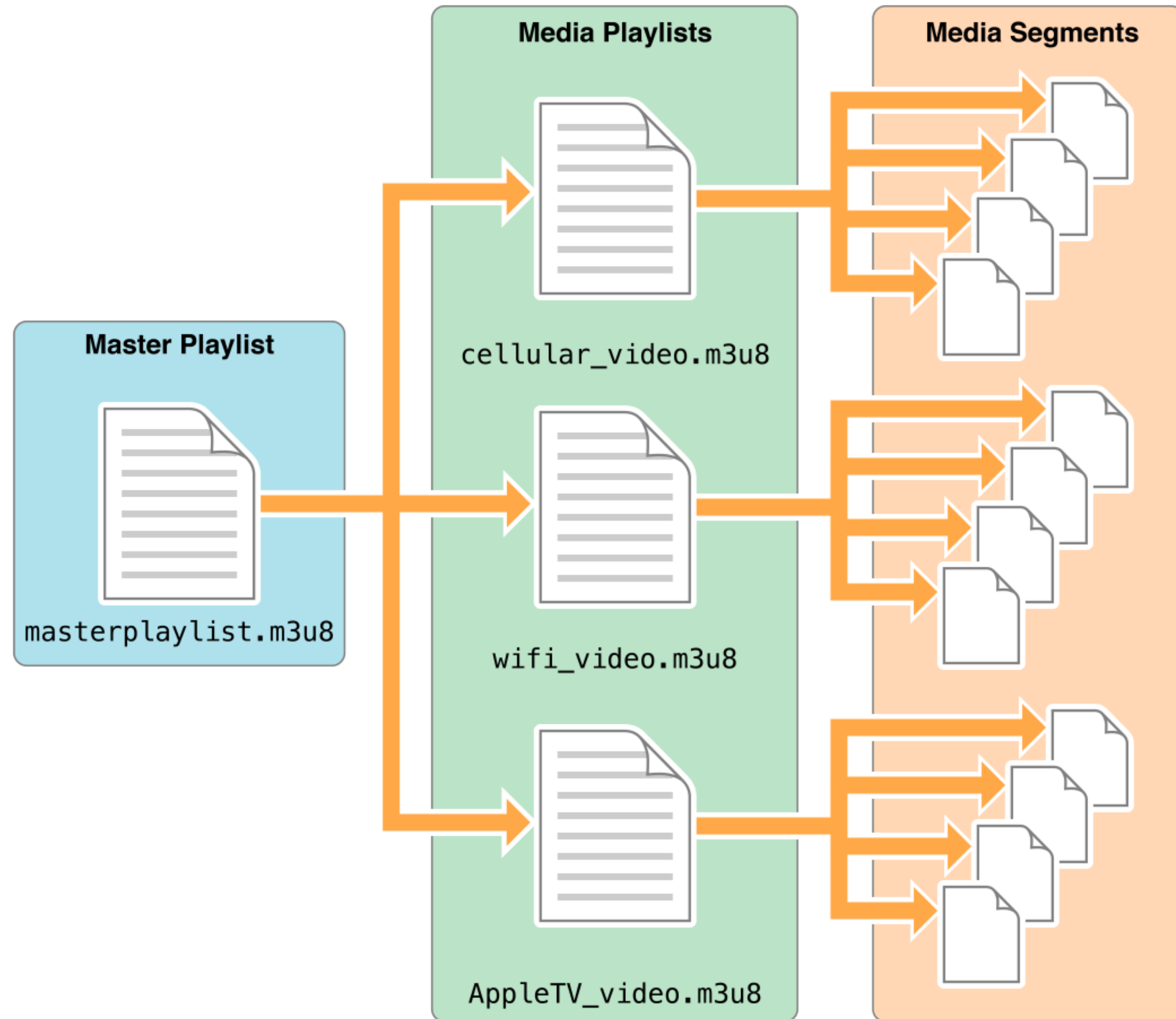
HLS 协议解析

→ 使用方式:

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

♥ HLS 协议解析

→ 基础架构:





HLS 协议解析

→ 基础架构:

```
curl http://qcloud.com/live/demo.m3u8
```

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:9.9001,
http://media.example.com/wifi/segment0.ts
#EXTINF:9.9001,
http://media.example.com/wifi/segment1.ts
```



HLS 协议解析

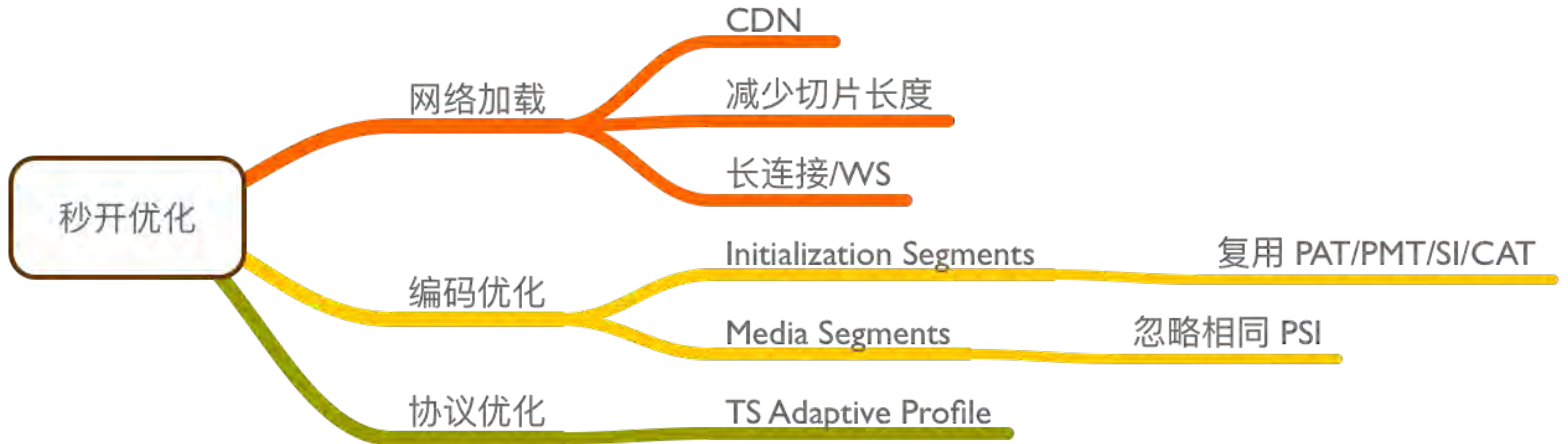
→ 秒开优化:

- 打开时延：页面加载到视频播放的时间差*
- 播放时延：采集端和播放端的时间差

```
#EXTINF:3.301,  
http://media.example.com/wifi/segment0.ts
```

♥ HLS 协议解析

→ 秒开优化:



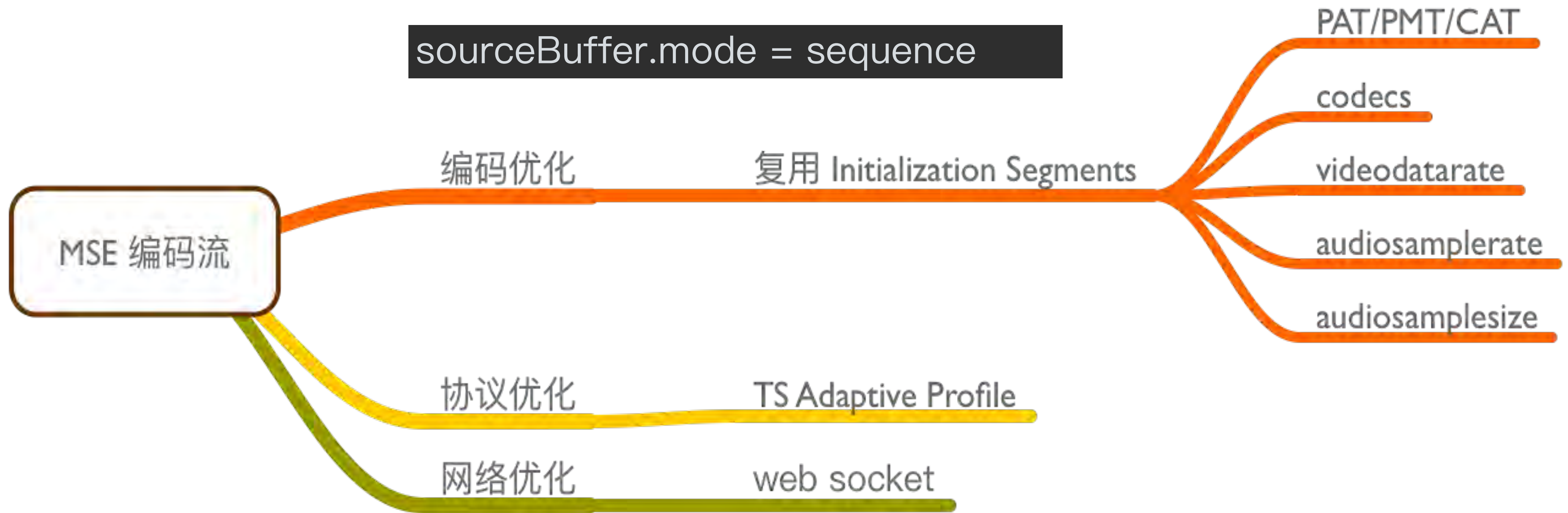
♥ HLS 协议解析

→ 秒开优化:



♥ HLS 协议解析

→ 秒开优化:





● MSE 提供播放流

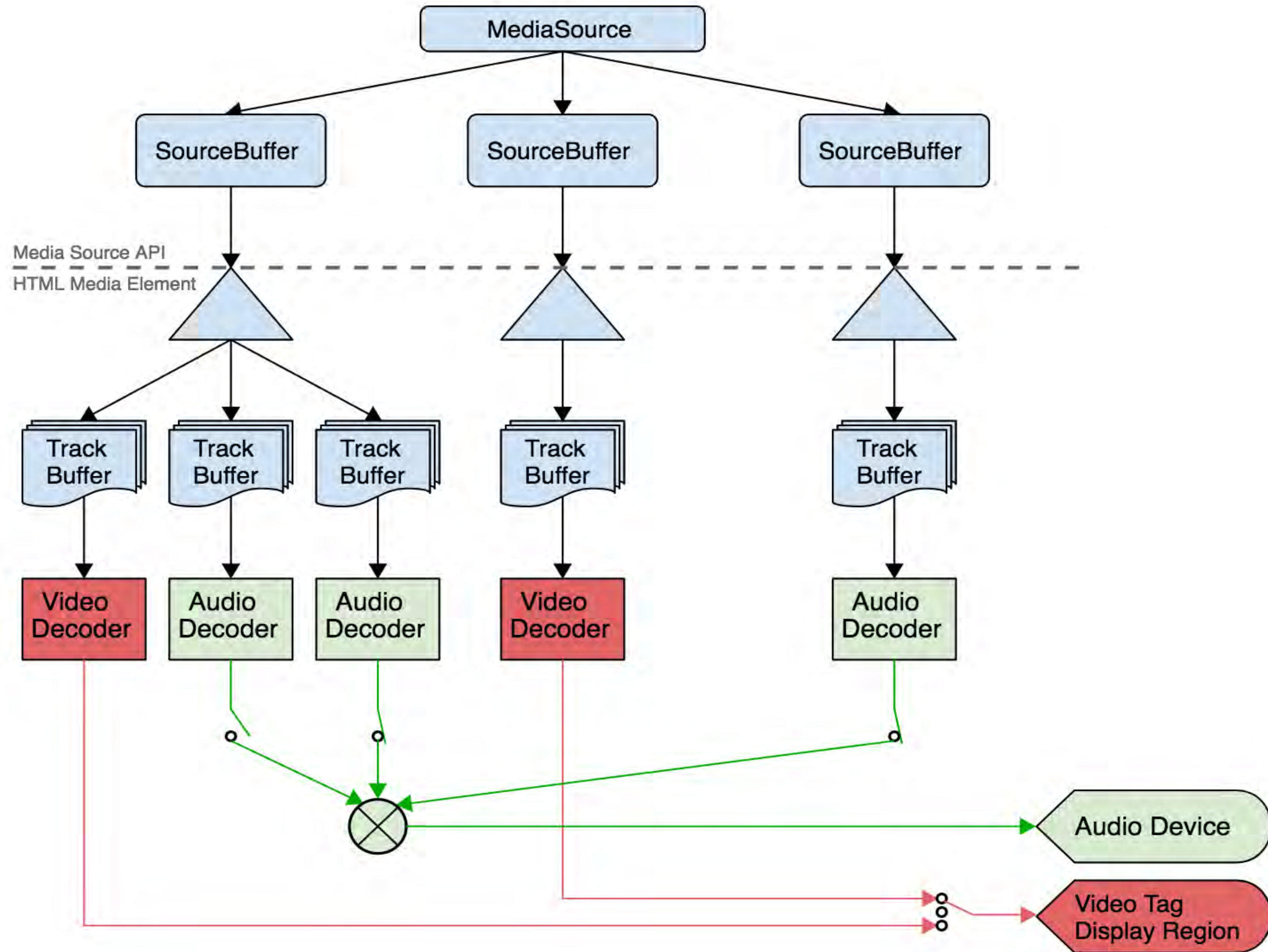


● Buffer 转换视频格式

🎯 直播技术实践

♥ MSE 提供播放流

→ MSE 架构



 MSE 提供播放流

→ MediaSource 与 video 连接

```
let mediaSource = new MediaSource();  
vidElement.src = URL.createObjectURL(mediaSource);
```

→ MediaSource 状态值

```
enum ReadyState {  
  "closed",  
  "open",  
  "ended"  
};
```

 MSE 提供播放流

→ SourceBuffer 提供视频流

```
MS.addEventListener('sourceopen',function(e){
var mediaSource = e.target;
var sourceBuffer = mediaSource.addSourceBuffer(mime);
vidElement.addEventListener('canplay',function(){
    if(vidElement.readyState >= 3){
        vidElement.play();
    }
},false);
if(!sourceBuffer.updating){
    sourceBuffer.appendBuffer(mediaBuffer);
}
},false);
```

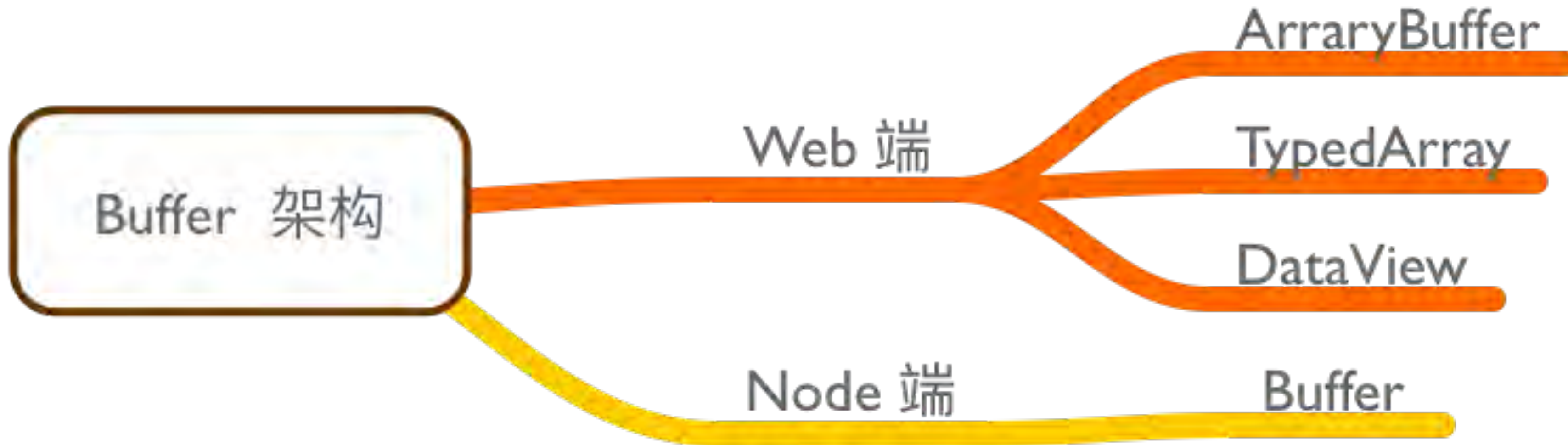
♡ Buffer 格式转换

→ Buffer 提供流的操作



♥ Buffer 格式转换

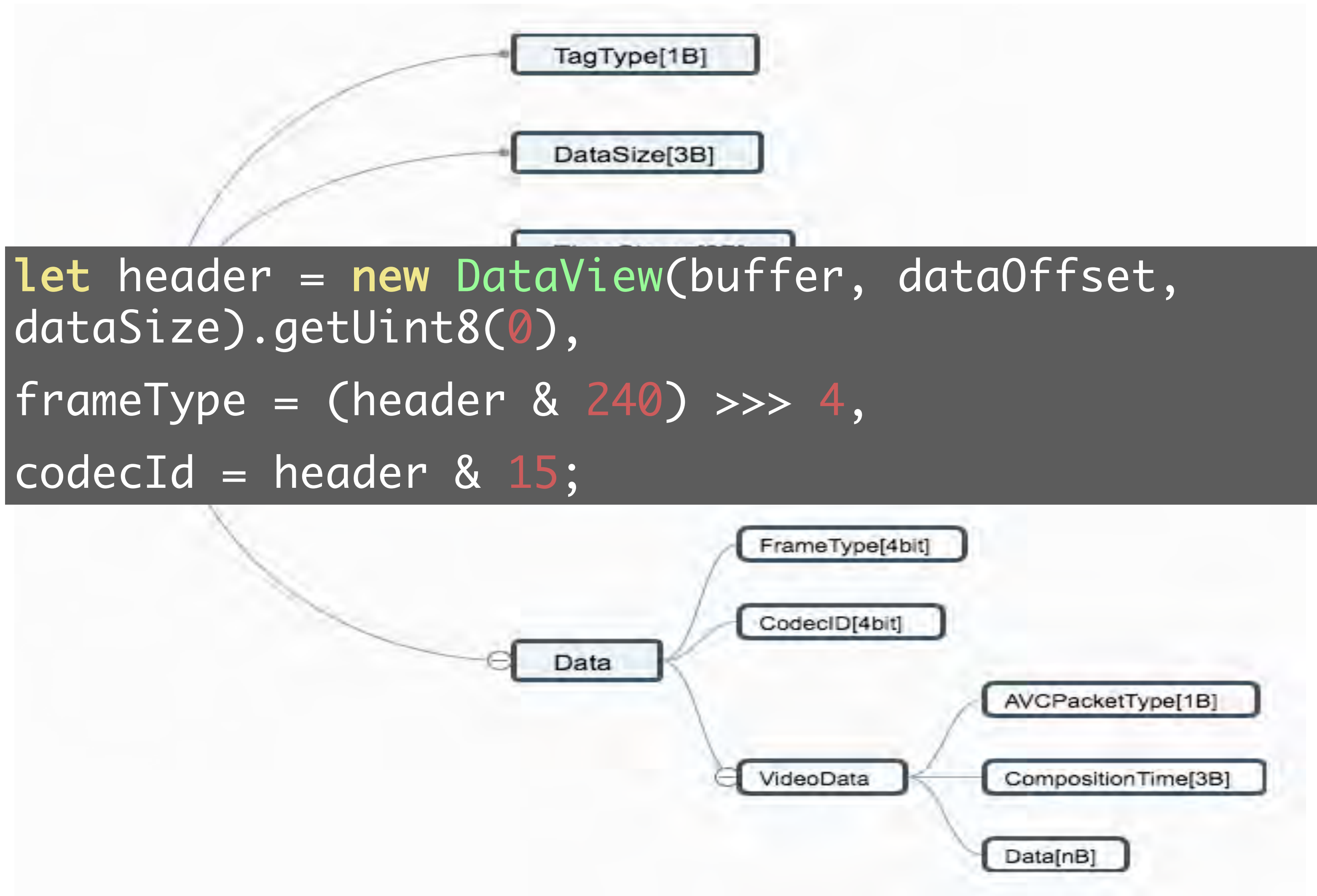
→ Buffer 编码流



♥ Buffer 格式转换

→ Buffer 编码流

```
let header = new DataView(buffer, dataOffset, dataSize).getUint8(0),  
frameType = (header & 240) >>> 4,  
codecId = header & 15;
```





● 流信息查看工具



● 报错信息的捕获



● 参考链接



🎯 H5 直播调试

 流信息查看工具

→ 字节查看

流信息的查看工具

```
var download = function (buffer) {
  let blob = new Blob(buffer, {
    type: 'application/octet-binary'
  });
  var url = window.URL.createObjectURL(blob);
  var a = document.createElement('a');
  a.style.display = 'none';
  a.href = url;
  a.download = 'test.mp4';
  document.body.appendChild(a);
  a.click();
  setTimeout(function () {
    document.body.removeChild(a);
    window.URL.revokeObjectURL(url);
  }, 100);
}
```



Buffer 格式转换

→ 报错信息的捕获

Player Properties

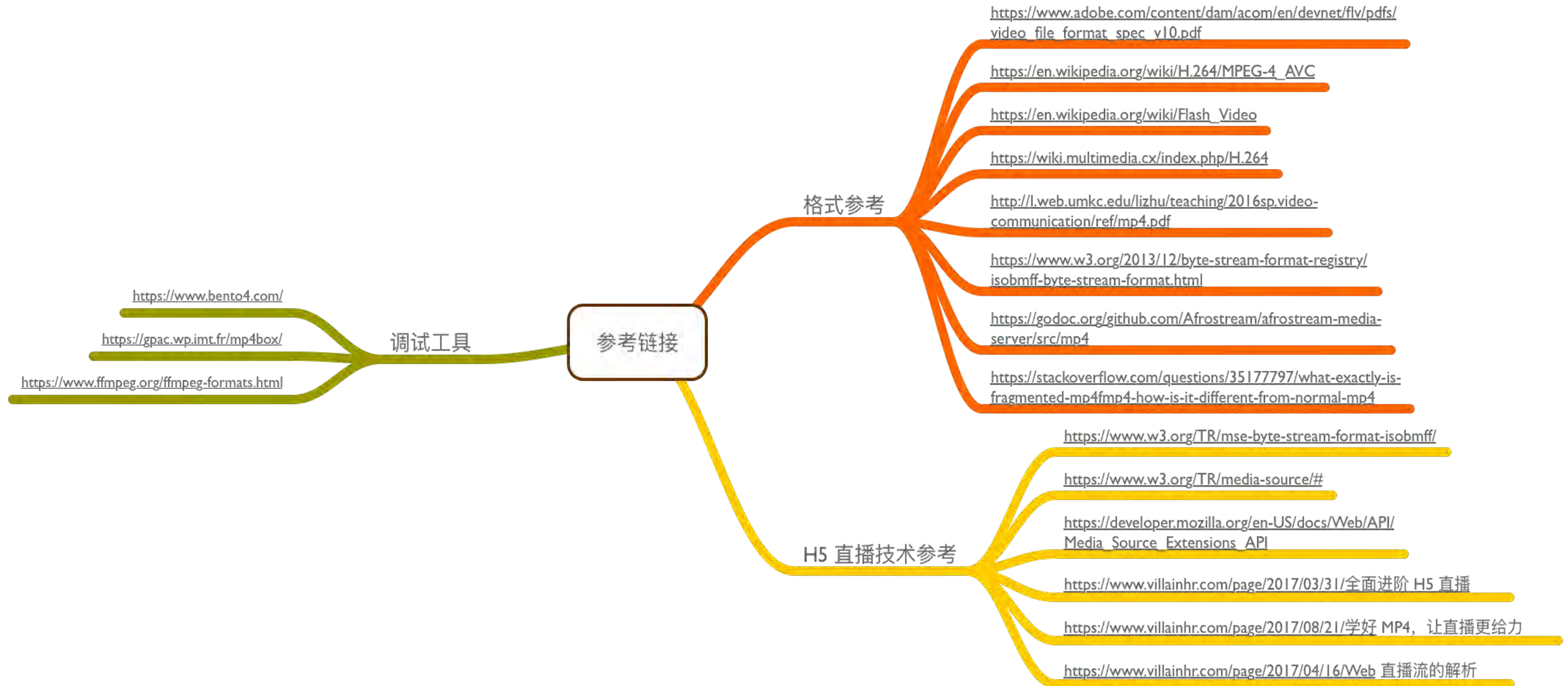
Copy to clipboard

报错

Property	Value
error	FFmpegDemuxer: open context failed
frame_title	NOW直播
frame_url	https://now.qq.com/h5/index.html?roomid=1210552999&_bid=2336&_wv=16778245
origin_url	https://now.qq.com/
passed_cors_access_check	false
pipeline_error	DEMUXER_ERROR_COULD_NOT_OPEN
pipeline_state	kStopped
player_id	7
range_header_supported	true
render_id	456
single_origin	true
streaming	false
total_bytes	639
url	http://6721.liveplay.myqcloud.com/live/6721_60339f7cb6260b648d5e1068d22a7d72.m3u8

ternals

♥ 参考链接



END

“

感谢您的聆听.

- 田淮仁