

MPEG-DASH 标准应用与实践

MPEG-DASH application and practice

bilibili – 丁建强 (Tuzkiss)

目录

I

MPEG-DASH 选型及理由

II

DASH 播放器实践及优化

III

标准思考及建议

首帧高

音画不同步

卡顿频繁

未来

成本高

切换体验不佳

音视频不独立

国际化

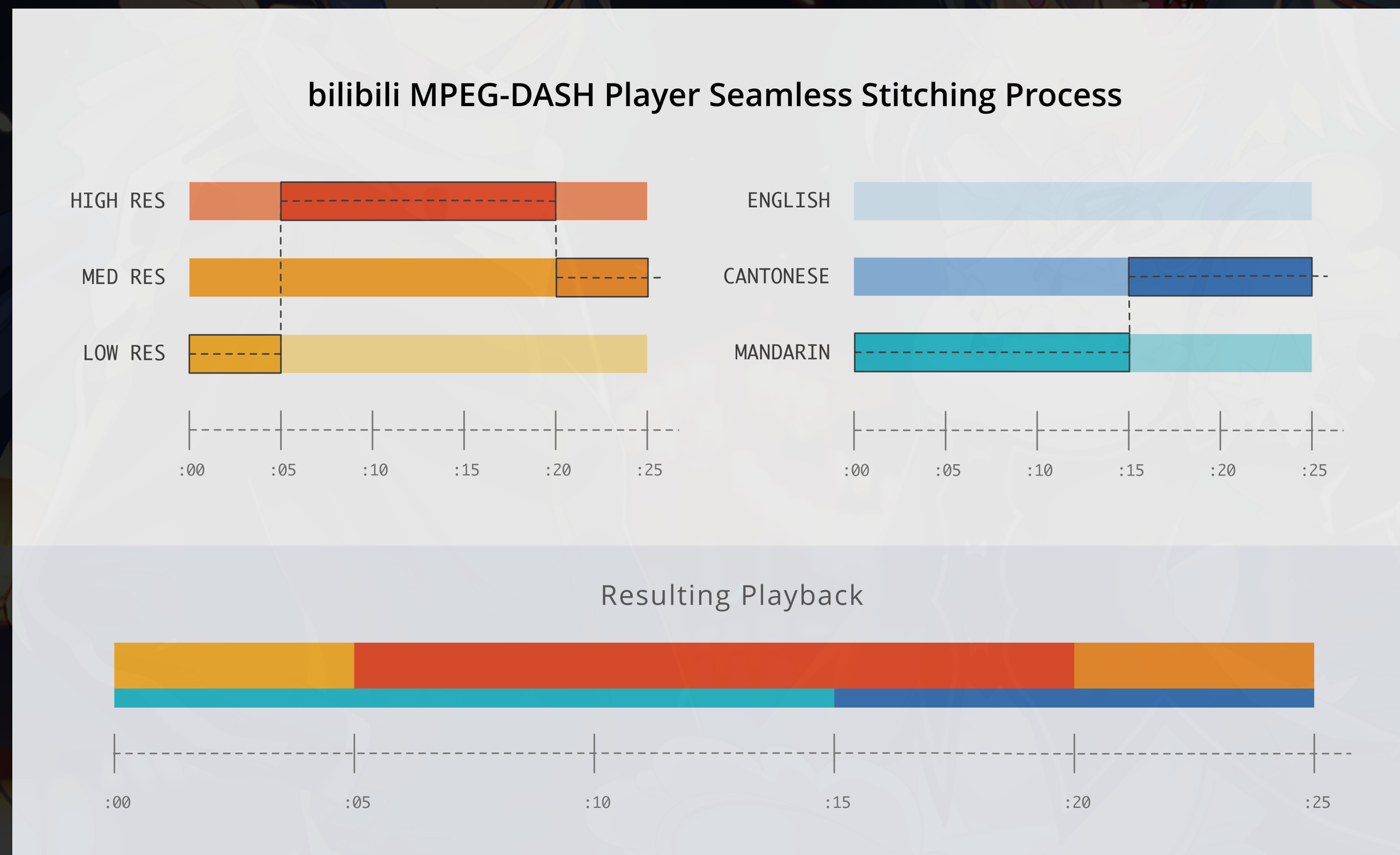
bilibili

简介

MPEG - Moving Picture
Experts Group

DASH - Dynamic Adaptive
Streaming over HTTP

MPEG-DASH 是一项通过通用
HTTP 服务器，高质量传输动态自
适应参数的媒体信息的技术。



格式	MPEG-DASH	HLS	FLV
标准类型	开放标准	苹果专有	Adobe专有(FLV)
切片格式	MPD	M3U8	Concat
切片编码	fMP4/WebM	MPEG2TS	FLV
视频编码	H264/H265 VP8/VP9	H264/H265	H264
音频编码	AAC MP3	AAC MP3	AAC MP3
浏览器支持	dash.js	hls.js	flv.js
音视频分离	支持	不支持	不支持
DRM	支持	部分支持	不支持
切片时长	灵活	固定	灵活
自适应码率	支持	支持	不支持
动态分辨率	支持	支持	不支持

业内成熟实践



B站企业文化支持

- 新技术尝试的支持

MPEG-DASH 播放器 实践及优化

► MPEG-DASH 播放器实践及优化 - 概览

Mime Type: video/mp4;codecs="avc1.64001E", audio/mp4;codecs="mp4a.40.2" [x]
Player Type: DashPlayer
Resolution: 1920 x 1080 [SAR 1:1]
FPS: 23.810
Video DataRate: 5690 Kbps
Audio DataRate: 311 Kbps
Segments: 10/593
Stream Type: https
Stream Host: cn-jszj-dx-v-06.acgvideo.com
Connection Speed: 12575 KB/s
Dropped Frames: 0/1060
Log: [Show] [Copy] [Download]

00:45 / 49:22 1080P+

- 2018.04 前期调研 & 数据支持
- 2018.06 本地 Demo & 定制开发
- 2018.08 基本功能 & 无缝切换
ABTest 灰度 & 自动清晰度
自动切换策略对比实验
音频自动清晰度
- 2018.09 全量 Dash 生产
埋点优化 & 完善错误处理
- ...

用户体验：

1、首帧速度

从页面打开到播放器载入首帧并开始播放时间。

2、卡顿率

从页面打开到播放器载入首帧并开始播放时间。

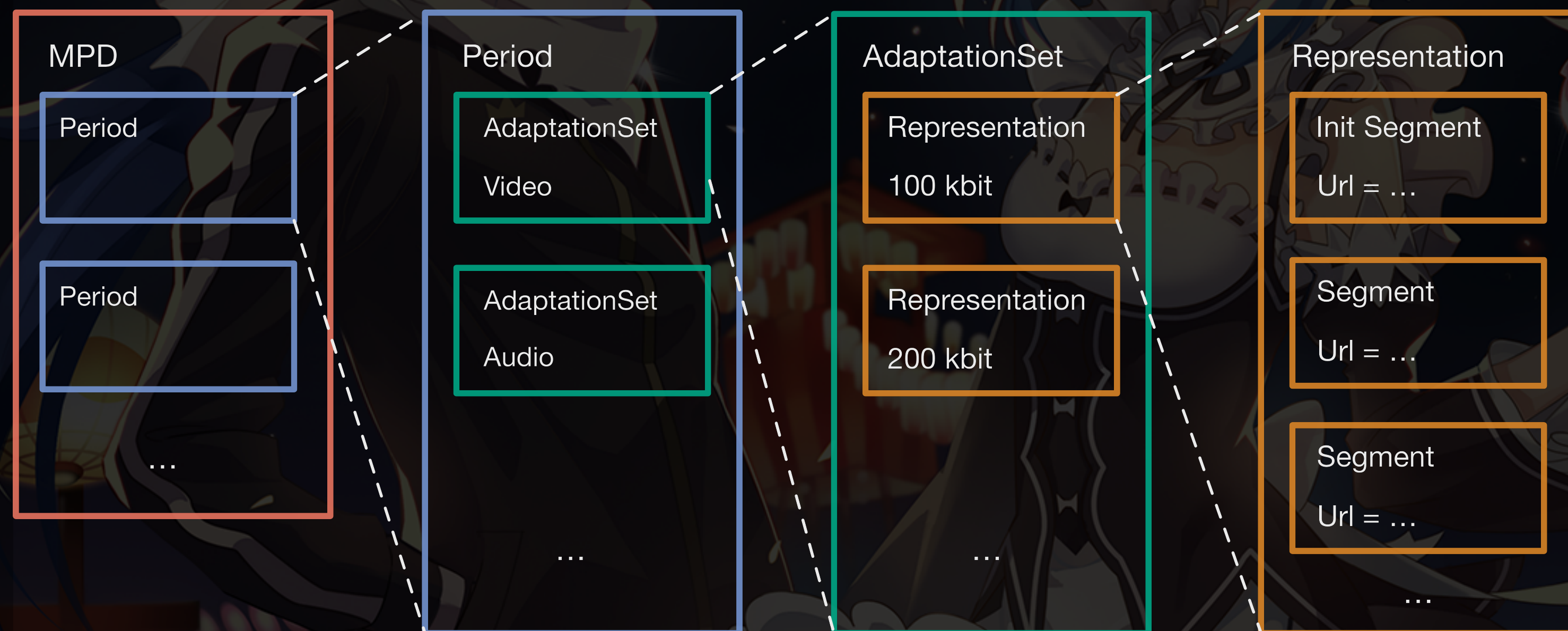
3、错误率

单次播放中遇到错误的概率。

首帧优化改造 - MPD 改造优化

MPD - Media Presentation Description

是一个描述整个 MPEG-DASH 码流的构成 XML Document, 通过 MPD 的内容可以构造出用于 HTTP GET 下载的 URL



首帧优化改造 - MPD 改造优化

MPD 文件存在冗余信息

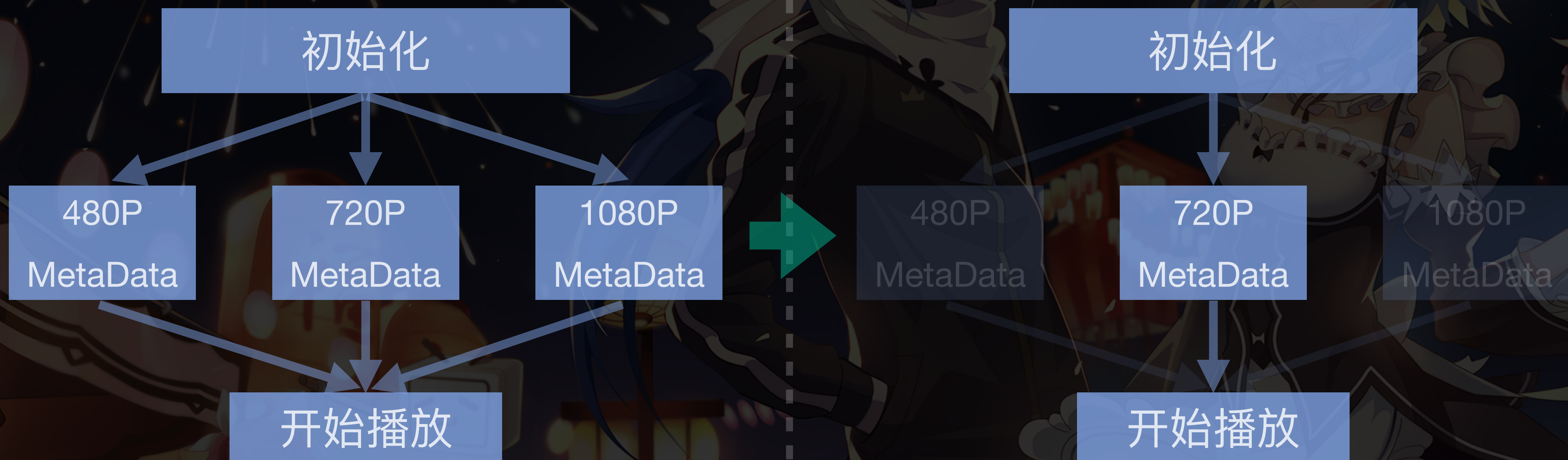
改造 & 优化

Manifest : JSON 描述关键信息

dash.js : 支持 JSON 信息解析

```
// Dash MPD => JSON
{
  "duration": 600,
  "minBufferTime": 1.5,
  "video": [{
    "id": "15",
    "mimeType": "video/mp4",
    "codecs": "avc1.64001E",
    "bandwidth": 922193,
    "width": 640,
    "height": 360,
    "frameRate": 25,
    "sar": "1:1",
    "startWithSAP": 1,
    "SegmentBase": {
      "Initialization": "0-987",
      "indexRange": "988-2483"
    },
    "baseUrl": "",
    "backupUrl": []
  }],
  "audio": [{
    "id": "30216",
    "mimeType": "audio/mp4",
    "codecs": "mp4a.40.2",
    "bandwidth": 67247,
    "SegmentBase": {
      "Initialization": "0-907",
      "indexRange": "908-2403"
    },
    "baseUrl": "",
    "backupUrl": []
  }]
}
```

清晰度初始化改造前后对比

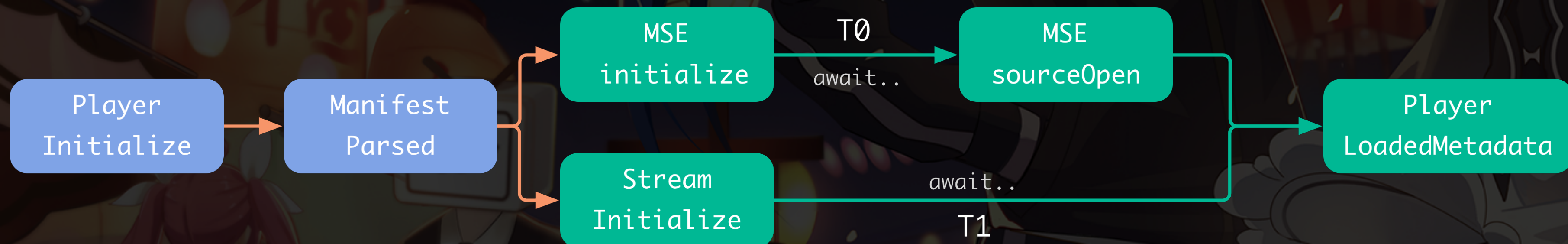


首帧优化改造 - MSE与拉流初始化优化

Before: $\text{Time} \approx T_0 + T_1$



After: $\text{Time} \approx \text{Max}(T_0, T_1)$



自适应码率切换策略对比实验

abrBola

- 基于当前缓冲区长度的切换策略

abrThroughput

- 基于当前网速的切换策略

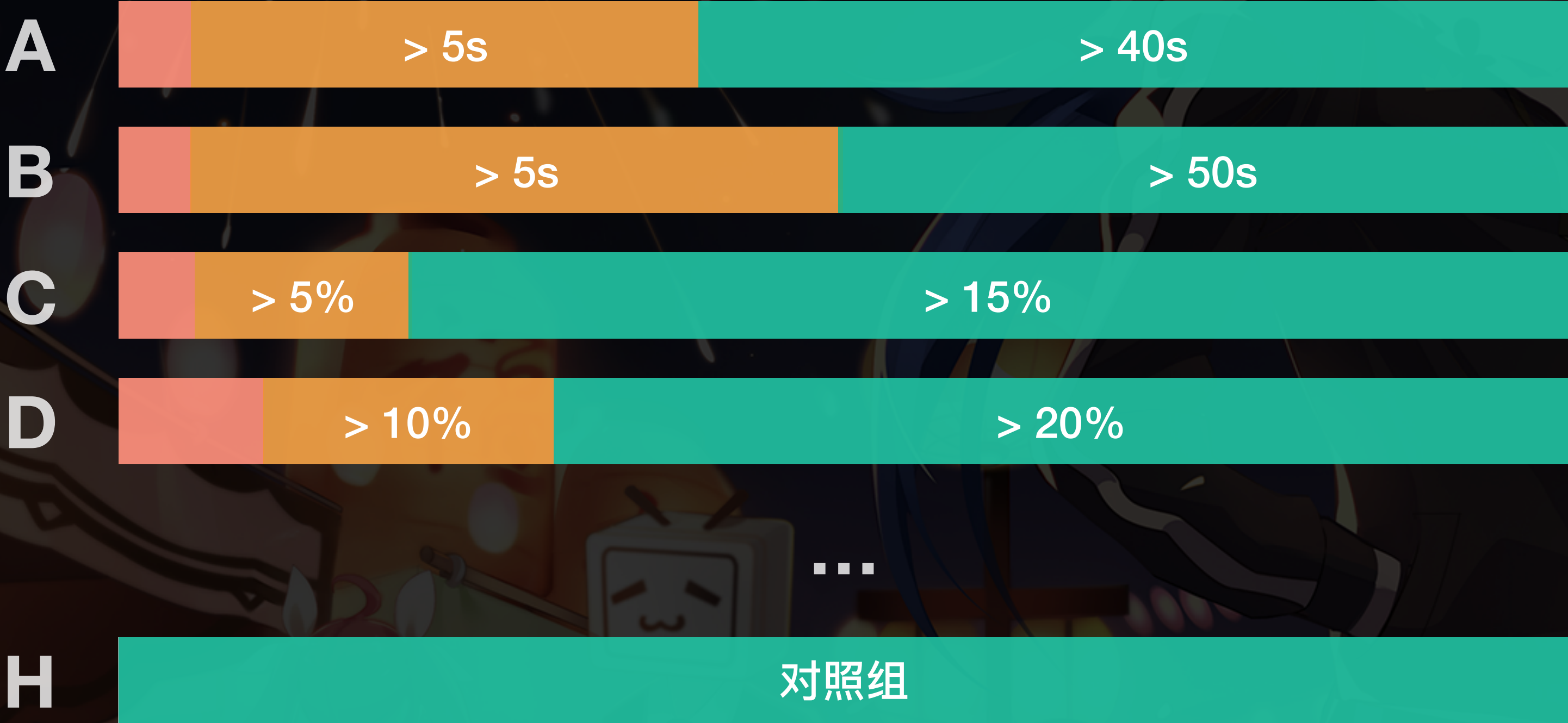
abrDynamic

- 自动在 abrBola 和 abrThroughput 间来回无缝切换，发挥两种策略的优势

日期	Session 尾号	卡顿次数	卡顿率	播放时长 (s)	使用策略	平均卡顿率
20180824	0	16166	3.31%	29317482	abrDynamic	3.30% ↓
	1	16105	3.29%	29393950		
	2	18683	3.78%	29623950	abrBola	3.67%
	3	17607	3.53%	29953112		
	4	18297	3.69%	29780312	abrThroughput	3.38%
	5	16888	3.45%	29349863		
	6	16051	3.27%	29432453		
	7	16679	3.42%	29272415		
20180825	0	22186	3.53%	37700163	abrDynamic	3.52% ↓
	1	22131	3.51%	37863477		
	2	23636	3.69%	38393555	abrBola	3.83%
	3	24273	3.86%	37776937		
	4	24888	3.95%	37772827	abrThroughput	3.55%
	5	21855	3.49%	37562931		
	6	22355	3.53%	38014339		
	7	22673	3.62%	37573116		
20180826	0	18254	3.16%	34668010	abrDynamic	3.32% ↓
	1	20232	3.49%	34816932		
	2	21707	3.70%	35188854	abrBola	3.61%
	3	20719	3.56%	34880971		
	4	20615	3.55%	34839817	abrThroughput	3.39%
	5	19867	3.44%	34695842		
	6	19546	3.35%	35024446		
	7	19785	3.38%	35171072		

abrDynamic > abrThroughput > abrBola

动态缓冲区实验



配置名	配置说明	配置颜色
StartBuffer	初始态的buffer长度	
T0	过渡到中间态的视频时间点	
Buffer0	中间态的buffer长度	
T1	过渡到最终态的视频时间点	
Buffer1	最终态的buffer长度	

错误率优化改造

错误异常

改造 & 优化

异常片段返回

异常打断 & 重试

网络抖动

超时处理

延迟超时处理

建联次数多

网络建联优化

I/O 存取优化

标准建议

播放体验 ∈ 用户观看体验

建议在 `HTMLVideoElement.getVideoPlaybackQuality()` 方法中新增属性标准，用于衡量部分用户观看体验。

firstFrameTime Read only

一个用毫秒表示从对象创建起开始到首帧渲染的

DOMHighResTimeStamp 对象。

bufferTimes Read only

一个表示 **HTMLVideoElement** 自创建起播放状态下发生卡顿次数的 unsigned long 值。

Properties

The `VideoPlaybackQuality` interface doesn't inherit properties from any other interfaces.

corruptedVideoFrames Read only

An unsigned long giving the number of video frames corrupted since the creation of the associated `HTMLVideoElement`. A corrupted frame may be created or dropped.

creationTime Read only

A `DOMHighResTimeStamp` containing the time in milliseconds between the start of the navigation and the creation of the object.

droppedVideoFrames Read only

An unsigned long giving the number of video frames dropped since the creation of the associated `HTMLVideoElement`.

totalVideoFrames Read only

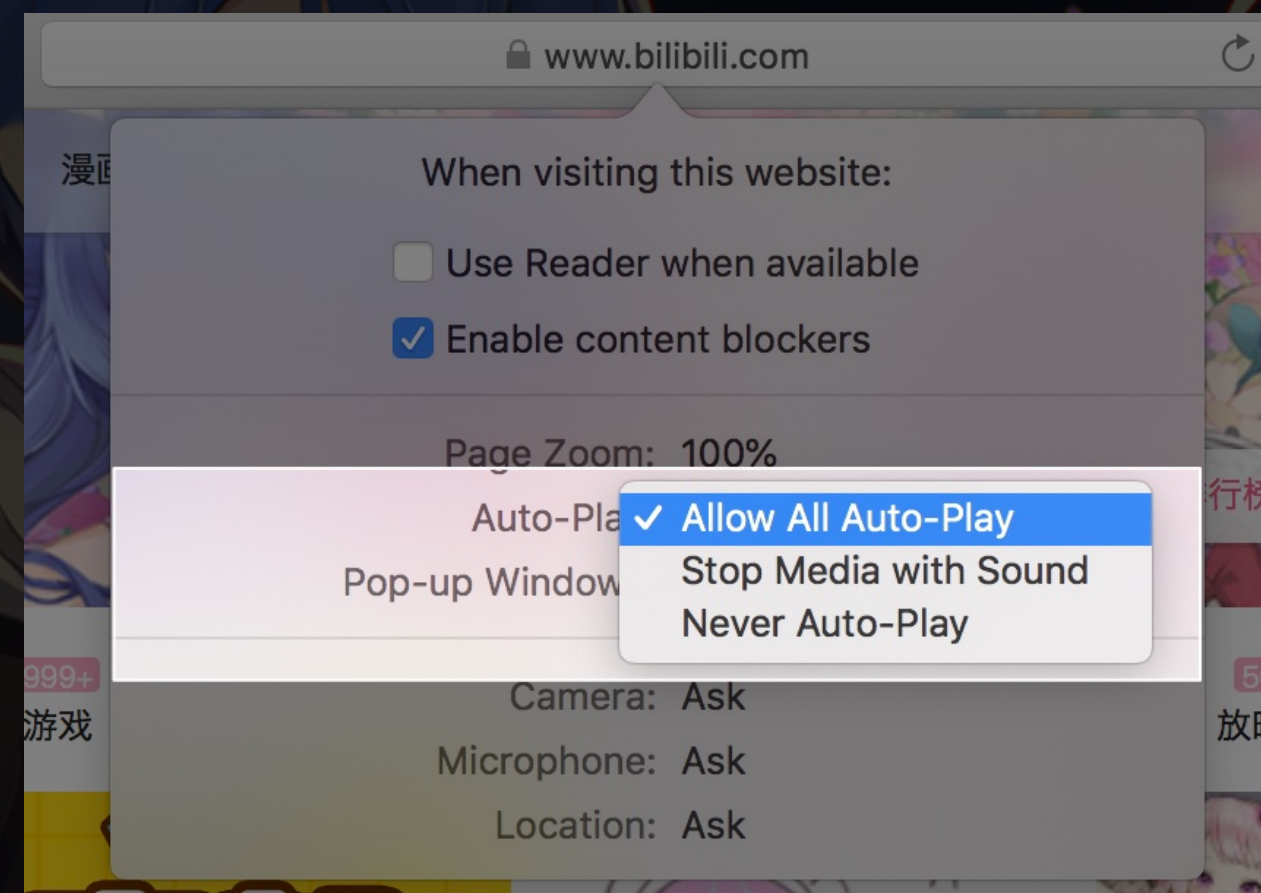
An unsigned long giving the number of video frames created and dropped since the creation of the associated `HTMLVideoElement`.

自动播放策略？ Autoplay Policy

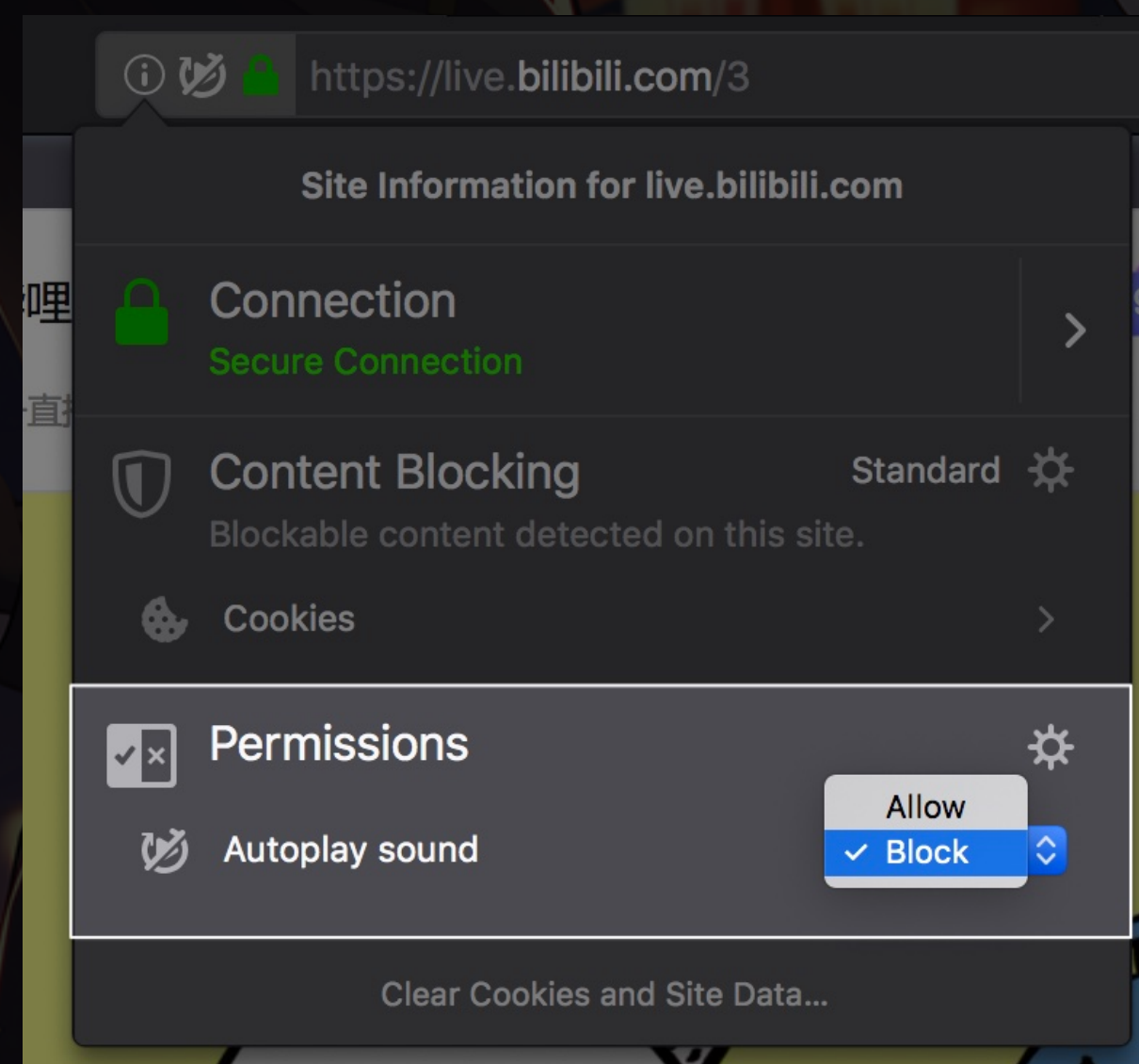
Media Engagement

Origin	Visits	Number of Playbacks	Last Playback	Is High	Score ▼
https://www.youtube.com/	14	4	2017-09-04T08:40:09.188Z	No	0.29
http://affinitynumerology.com/	1	0		No	0
http://caniuse.com/	1	0		No	0
http://developer.samsung.com/	1	0		No	0
http://javascriptweekly.com/	1	0		No	0
http://localhost:4000/	1	0		No	0
http://localhost:8080/	2	0		No	0
http://mathworld.wolfram.com/	1	0		No	0
http://numbermatics.com/	2	0		No	0
http://videosciences.cafe-sciences.org/	1	0		No	0
http://www.androidauthority.com/	1	0		No	0
http://www.escapistmagazine.com/	1	0		No	0
http://www.frandroid.com/	1	0		No	0
http://www.jilion.com/	1	0		No	0

 **Chrome 66 +**



 **Safari 11 +**

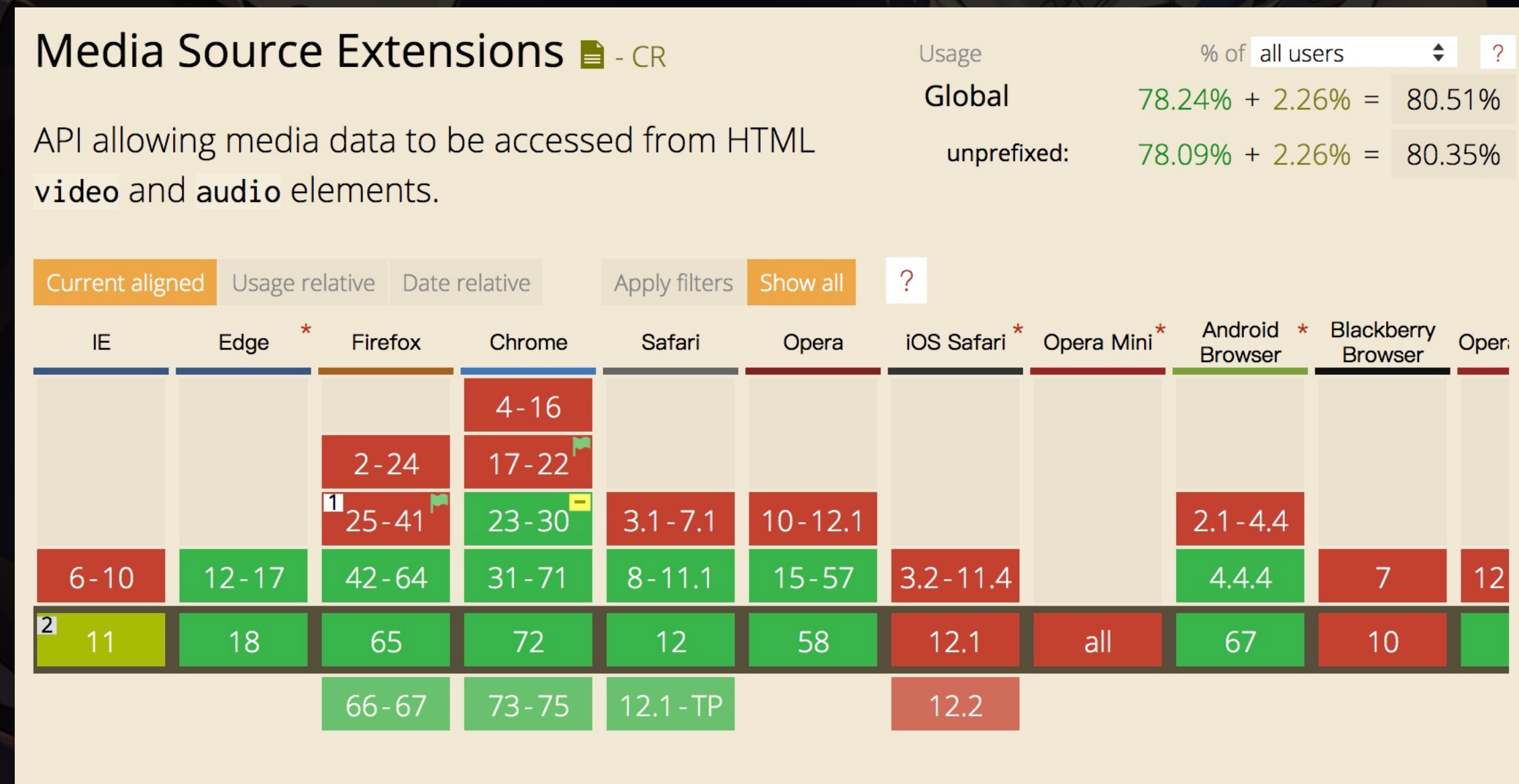


 **Firefox 66 +**

bilibili

其它

- 希望获取当前总共可使用 MSE 内存上限
- 希望推进 MSE 标准在移动端的完整支持
- 希望推进 Fetch 标准在各浏览器的统一化



Thanks

- Tuzkiss

Q & A