

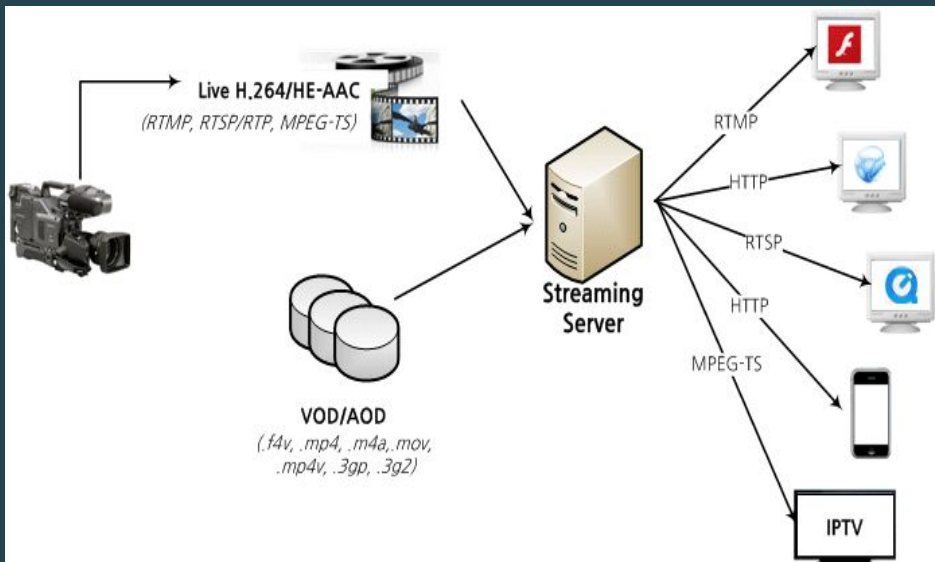
# HLS

## HTTP Live Streaming

# HLS란

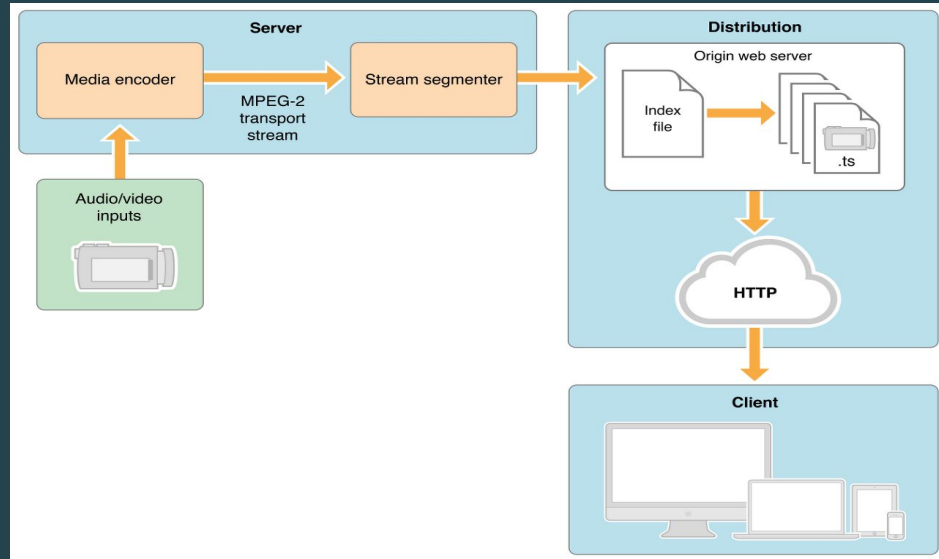
- 애플이 개발하여 2009년 출시한 HTTP 기반 적응 비트레이트 스트리밍 통신 프로토콜
- 웹 브라우저, 모바일 기기, 스트리밍 미디어 서버에서 지원되고 있는 가장 대중적인 스트리밍 포맷
- 특징
  - 네트워크 상태에 따라 영상 화질을 선택할 수 있는 Adaptive Bitrate Streaming 기능 제공
  - 사용자가 영상 재생 시 전체 영상 파일을 다운받지 않아도 재생 가능
  - 특정 재생 위치부터 영상을 보는 경우 해당 위치부터 영상 다운로드 가능

# Why HLS?



## HLS 이전의 스트리밍 방식

- 플레이어가 스트리밍 서버에 연결한 스트리밍 규격에 맞게 다시 데이터를 변형하여 전송한다
- 웹 서버에 비하여 처리 과정이 더 많기 때문에 비용이 많이 든다

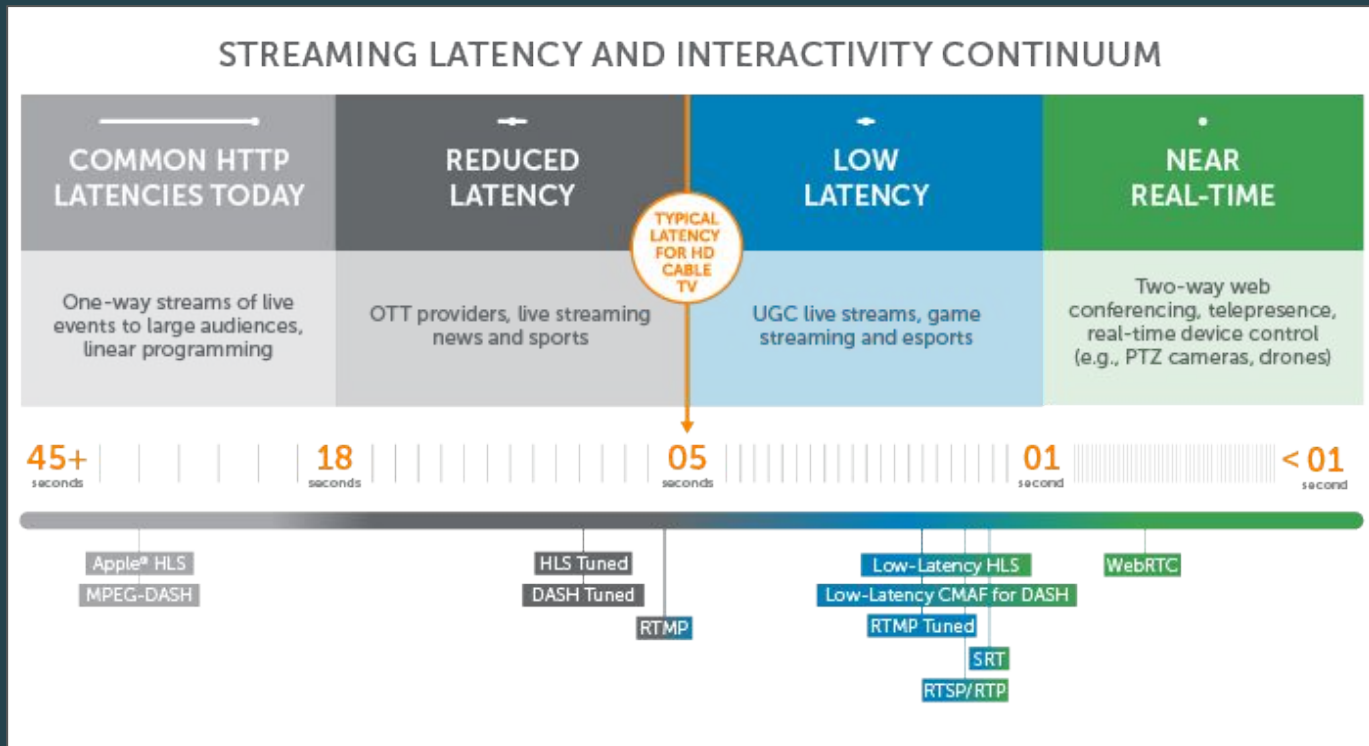


## HLS의 스트리밍 방식

- 미디어 서버는 MPEG-TS 로 인코딩 및 조각화한다
- 분산 서버는 HTTP로 요청을 받아서 플레이어에 응답을 주는 역할만 한다
  - 요청받은 파일을 읽어서 어떠한 변형도 하지 않고, 읽은 그대로 응답으로 보내기만 하기 때문에 어떤 웹서버든 사용 가능하다

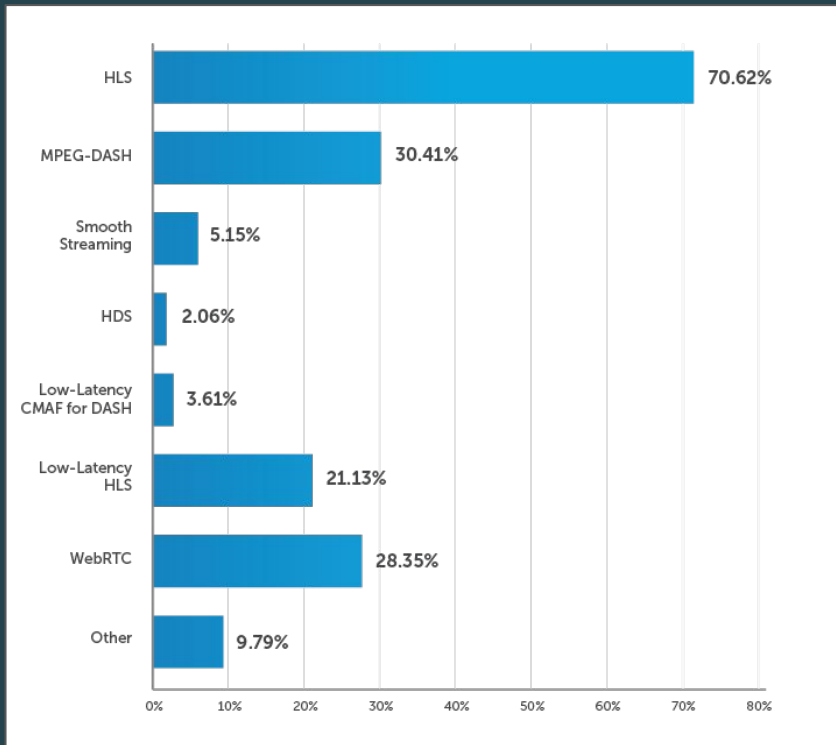
# Why HLS?

## Streaming 방식에 따른 지연시간



# Why HLS?

2021 사용중인 스트리밍 방식 점유율



# Why HLS?

- 규격 자체가 단순하여 손쉽게 HLS 스트리밍 프로토콜 구현이 가능하다
  - 쉽게 HLS를 구현이 가능하여 Apple외의 다른 많은 업체들도 HLS를 사용하게 되었다
- Apple기기의 사용자 수가 늘어나면서 자연스럽게 수요와 영향력이 늘어나게 되었고 현재 가장 대중적인 스트리밍 프로토콜이다
- HTTP 데이터를 전송하므로, RTP 또는 RTMP와 같은 전통적인 스트리밍 프로토콜을 다양하게 개선한다
  - 간편한 HTML5 플레이어 구현 가능
  - CDN과 기타 HTTP 캐싱 인프라에서 캐시 기능 사용 가능
  - 가변 비트레이트 스트리밍 기능을 통한 클라이언트 최적화
  - 간편한 이중화 기능
  - 인프라 비용 절감
  - 프록시와 방화벽 제한을 통해 위협 저감

# HLS Workflow

HLS는 분할된 MPEG-TS 비디오와 m3u8 설명자 파일을 사용하여 가변 비트레이트 라이브 비디오와 주문형 비디오를 제공하는 기술로 영상을 품질별로 인코딩 및 조각화하여 저장한다. 영상은 .ts 확장자로 저장한다

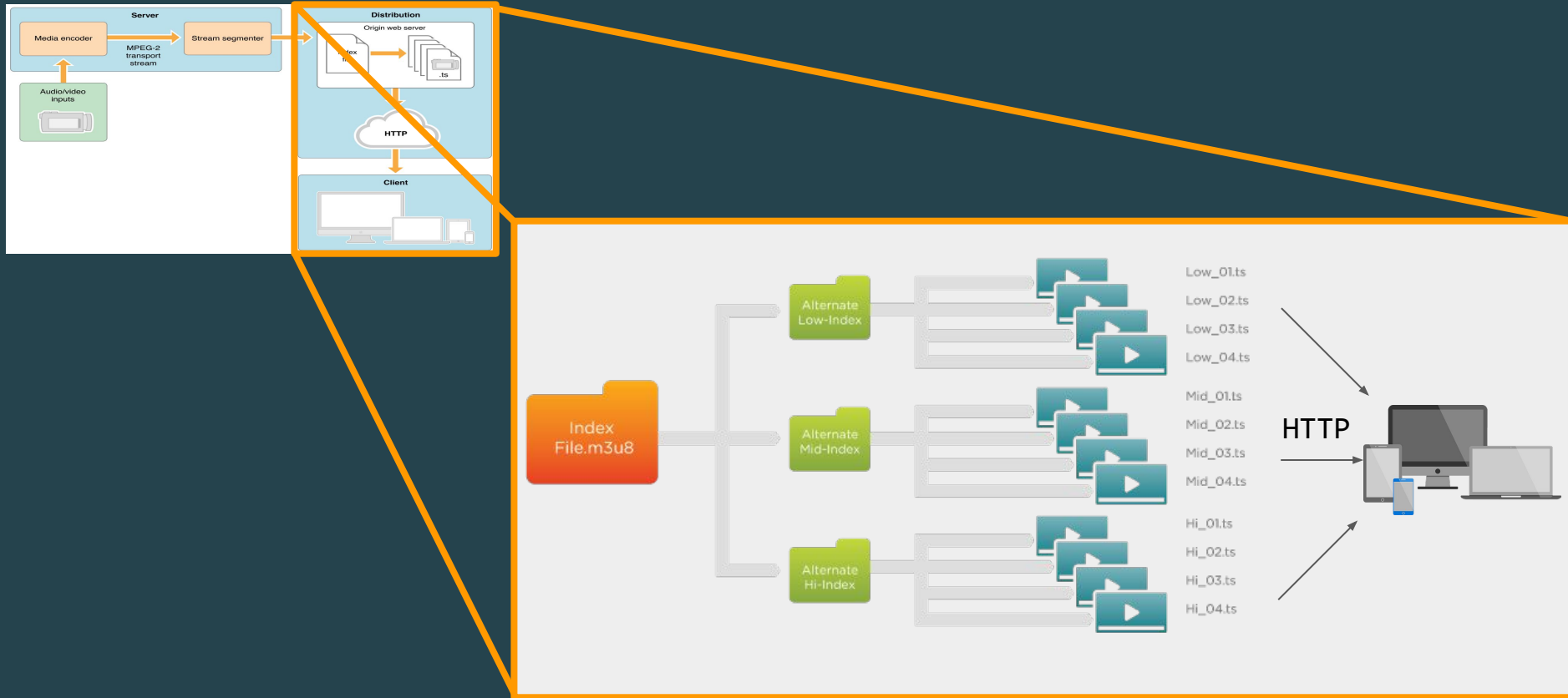
- 인코딩: 비디오 데이터의 포맷을 다시 설정하여 모든 장치가 데이터를 인식하고 해석할 수 있게 한다.
  - HLS 은 H.264나 H.265 인코딩을 사용한다.
- 조각화: 비디오는 몇 초 길이의 세그먼트로 나눈다. 세그먼트 길이는 다양하지만 기본 길이는 6초이다 (2016년까지는 10초)
  - 비디오를 세그먼트로 나누는 것과 더불어 HLS는 비디오 세그먼트의 인덱스 파일을 만들어 세그먼트의 순서를 기록한다.
  - HLS는 또한 480p, 720p, 1080p 등의 다양한 품질로 여러 세트의 세그먼트를 복제한다.

HLS는 m3u8 파일과 MPEG-TS 영상 파일로 구성되어 있다.

- m3u8(매니페스트) 파일 : 다른 m3u8 파일이나 동영상 세그먼트 (MPEG-TS)를 참조한다
- MPEG-TS : 인코딩 및 조각화된 동영상 파일

플레이어는 m3u8파일을 읽어서 다양한 화질의 동영상을 받을 수 있으며, 자신의 네트워크 상태에 따라 원하는 동영상 품질을 다운받고 순차적으로 재생하는 방식으로 가변 비트레이트 스트리밍 (Adaptive Bitrate Streaming) 기능을 사용할 수 있다

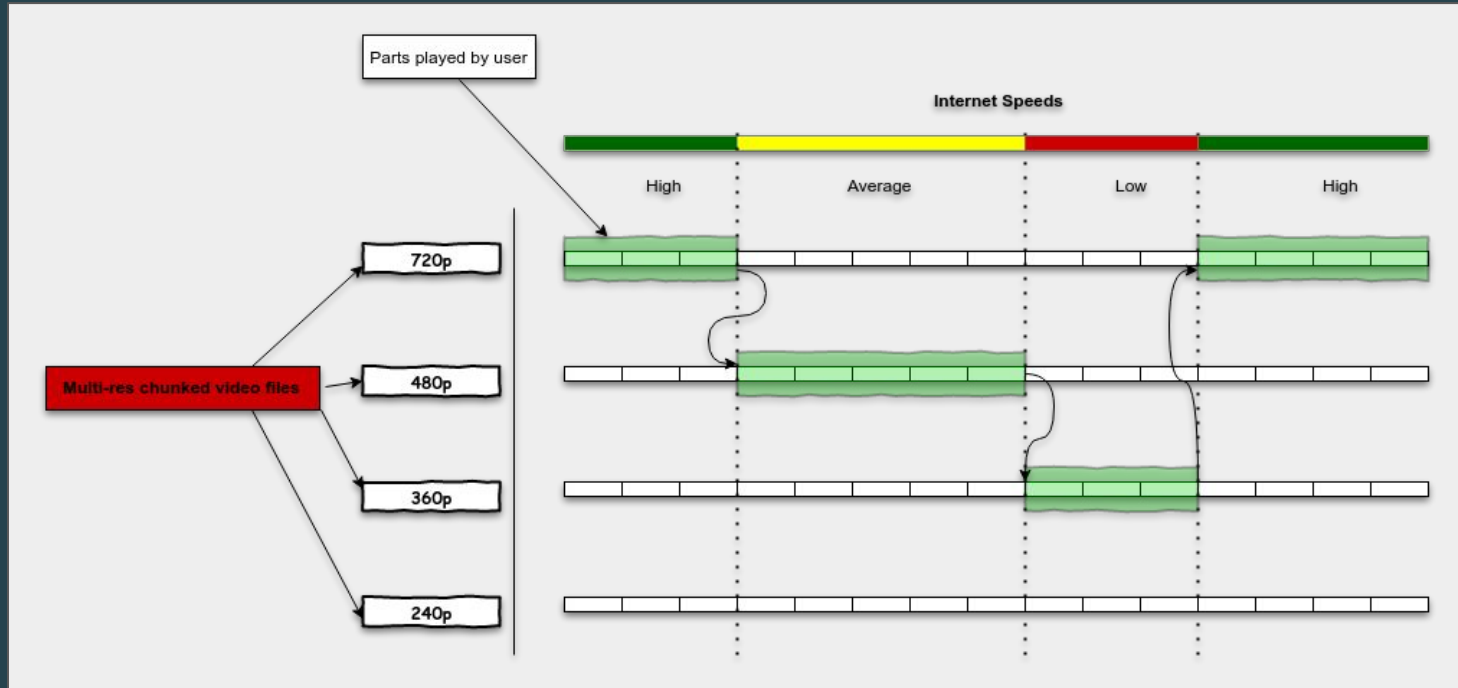
# HLS Workflow





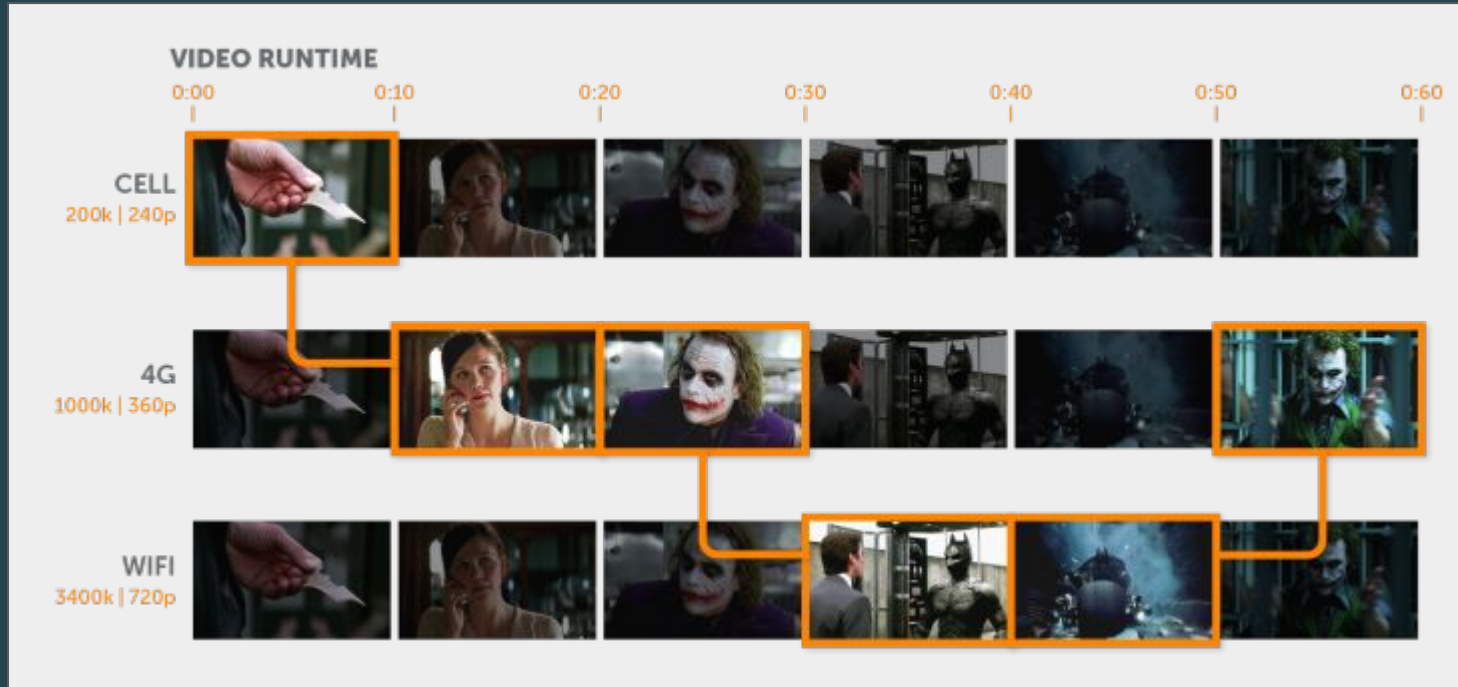
# HLS Workflow

## HLS Adaptive Bitrate Streaming



# HLS Workflow

## HLS Adaptive Bitrate Streaming



# HLS Workflow

## m3u8 & MPEG-2 TS 파일

```
#EXTM3U
#EXT-X-TARGETDURATION:2
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:2,
segment-000000.ts
#EXTINF:2,
segment-000001.ts
#EXTINF:2,
segment-000002.ts
```

segment-000000.ts

segment-000001.ts

segment-000002.ts

### m3u8

연속으로 재생할 파일 목록을 가진 플레이 리스트 파일 각 줄마다 재생할 파일의 경로와 여러 가지 지시어로 재생할 파일에 대한 추가적인 정보를 제공할 수도 있다

### MPEG-2 TS

오디오나 비디오, 방송의 채널 정보 등을 전송하거나 저장하기 위해 정의한 규격

# HLS Workflow

## m3u8 파일의 지시어

지시어	형식	역할
#EXTM3U	#EXTM3U	파일의 가장 첫 줄에 명시하여 파일이 m3u8 포맷임을 명시한다.
#EXTINF	#EXTINF: <재생 시간:초>.<제목>	이 지시어의 다음에 명시된 콘텐츠의 재생 시간과 제목을 명시한다.
#EXT-X-TARGETDURATION	#EXT-X-TARGETDURATION: <시간: 초>	파일 목록에 나열된 각 파일의 최대 재생 시간을 명시한다.
#EXT-X-ENDLIST	#EXT-X-ENDLIST	플레이 리스트에서 재생할 콘텐츠가 더 이상 없음을 의미. 이 지시어가 표시된 줄 이후의 콘텐츠는 무시한다.
#EXT-X-DISCONTINUITY	#EXT-X-DISCONTINUITY	이 지시어가 표시된 줄을 기준으로 이전 줄과 이후 줄에서 재생하는 콘텐츠의 정보가 변경되었음을 표시. 예를 들어 이전 콘텐츠와 이후 콘텐츠의 파일 포맷, 파일이 갖고 있는 미디어 트랙의 개수, 인코딩 정보, 재생 시간 정보 등이 변경되면 이 지시어를 플레이리스트에서 정보가 바뀌는 파일 사이에 명시하여 플레이어가 새로운 정보를 사용해야 하는 시점을 알려 준다.
#EXT-X-MEDIA-SEQUENCE	#EXT-X-MEDIA-SEQUENCE: <첫 파일의 일련번호>	제일 먼저 플레이해야하는 파일의 일련번호를 명시한다.
#EXT-X-KEY	#EXT-X-KEY: <암호화 방법>[, <key>]	암호화된 파일을 해독하는 키 값을 명시한다. HTTP Live Streaming에서는 재생 시간에 따라 분할된 각 파일을 암호화하여 전송할 수 있다. 암호화된 파일을 해독할 때 필요한 키 값을 플레이어에게 알려 주기 위해 사용한다.
#EXT-X-STREAM-INF	#EXT-X-STREAM-INF	이 지시어 다음 줄의 콘텐츠에 대한 정보를 제공한다. #EXTINF는 재생 시간에 대한 정보만 제공하고, #EXT-X-STREAM-INF는 다음과 같은 정보를 제공한다. <ul style="list-style-type: none"><li>BANDWIDTH: 10진수로 표시한 bps 값</li><li>PROGRAM-ID: 플레이 리스트 파일에 있는 콘텐츠가 갖는 고유 값</li><li>CODEC: 해당 콘텐츠에 적용된 코덱(codec) 정보</li><li>RESOLUTION: 해상도</li></ul>

# HLS Workflow

## MPEG-2 TS

- HLS에서 사용하기 위한 .ts 파일은 MPEG-2 TS를 순서대로 저장한 파일이다. 다만 정한 시간에 따라 분할하여 저장한다
- HLS로 화면을 자연스럽게 재생하려면 각 ts 파일이 I-frame(Intra frame: 화면 전체가 압축되어 들어 있는 frame)을 포함해야 한다. 되도록이면 첫 비디오 데이터가 I-frame인 것이 좋다

## Apple의 미디어 규격 권고 사항

	비디오	오디오
코덱	<ul style="list-style-type: none"><li>● iPhone 3G / iPod 2세대 이상이면 H.264 Baseline 3.1 이하</li><li>● 이전 버전의 iPhone / iPod이면 H.264 Baseline 3.0 이하</li><li>● iPad, Apple TV2, iPhone 4 이상이면 Baseline profile 3.0/3.1 혹은 Main profile 3.1 이하</li></ul>	<ul style="list-style-type: none"><li>● iPhone 3G / iPod 2세대 이상 ? H.264 Baseline 3.1 이하</li><li>● HE-AAC 혹은 AAC-LC, 스테레오</li><li>● MP3, 스테레오</li></ul>
프레임 레이트 비트 레이트	<ul style="list-style-type: none"><li>● 200 kbps 미만: 10 fps</li><li>● 200~300 kbps: 12 ~ 15 fps</li><li>● 기타: 29.97 fps</li></ul>	<ul style="list-style-type: none"><li>● 샘플링 레이트: 22.05 khz</li></ul>

# HLS vs MPEG-DASH

- MPEG-DASH는 HLS와 마찬가지로 비디오를 작은 조각으로 나누고 다양한 품질 수준에서 인코딩하여 스트리밍 한다는 점에서 HLS와 매우 유사하며, HLS의 가장 강력한 경쟁자다
  - DASH도 HLS와 마찬가지로 인코딩 및 조각화 과정을 사용하여 적응 비트 전송률 스트리밍을 지원한다

## HLS / DASH 주요 차이

	HLS	MPEG-DASH
인코딩	<ul style="list-style-type: none"><li>● 과거 : H.264</li><li>● 현재 : H.264, H.265 코덱 사용</li></ul>	<ul style="list-style-type: none"><li>● 코덱에 구애받지 않고 어떠한 인코딩 표준도 사용할 수 있다</li></ul>
매니페스트	<ul style="list-style-type: none"><li>● m3u8 파일</li></ul>	<ul style="list-style-type: none"><li>● mpd 파일</li></ul>
품질	<ul style="list-style-type: none"><li>● 과거 : H.264밖에 지원하지 않아서 고해상도 비디오 지원이 불가능</li><li>● 현재 : HEVC/H.265를 지원하며 고해상도 지원 가능</li></ul>	<ul style="list-style-type: none"><li>● 코덱에 구애받지 않기 때문에 원하는대로의 품질 제공이 가능하다</li></ul>
세그먼트 길이	<ul style="list-style-type: none"><li>● 2016년 이전에는 기본 세그먼트 길이가 10초였지만 현재는 기본길이가 6초이며 조절이 가능하다</li></ul>	<ul style="list-style-type: none"><li>● 세그먼트의 통상 길이가 2 ~ 10초이다</li><li>● 최적 세그먼트 길이는 2 ~ 4초이다</li></ul>
호환성	<ul style="list-style-type: none"><li>● iOS는 물론 Android, Mac, Linux 및 Windows와 같은 모든 최신 운영 체제 또는 장치는 HLS를 지원한다</li></ul>	<ul style="list-style-type: none"><li>● MSE(Media Source Extensions)를 지원하는 모든 기기에서 호환된다</li><li>● 모바일 Safari에서는 MSE를 지원하지 않아 사용 불가능하다</li></ul>

# 결론

- 모바일 기기가 확대될수록 HTTP를 이용한 스트리밍 방식이 기존 방식의 스트리밍 서비스를 대체할 것이다. HLS는 온라인 비디오를 제공하기 위해 사용되는 프로토콜 중 가장 광범위하게 사용중인 프로토콜이다
- MPEG-DASH와 HLS는 기술적으로 유사하며, MPEG-DASH가 HLS보다 조금 더 진보된 기술을 제공하지만 Apple기기에서의 호환성 문제 때문에 HLS가 더 많이 쓰인다
  - 세계적으로 10억개 이상의 Apple 장치가 있고 이는 무시할만한 수치가 아니다
- 가장 광범위한 호환성을 원한다면 HLS를 선택하는 것이 좋다
  - Safari가 MPEG-DASH를 지원할 경우는 얘기가 달라진다
- HLS보다 더 높은 호환성을 원한다면, HLS / MPEG-DASH 둘 다 사용하는 방법이 있다
  - HLS용 영상과 DASH 영상을 저장해야 하므로 필요한 용량은 두배로 늘어난다
  - HLS, DASH, Smooth, HDS등의 스트리밍 규격을 통합하려는 시도가 진행중이며 미래에는 통합된 기술을 사용할 수도 있다
- Low Latency HLS라도 다소의 지연시간이 있기 때문에 대화나 드론 카메라 등 1초 미만의 스트리밍이 중요할 때 WebRTC를 사용하는 것이 좋고, 그 외에는 HLS의 사용이 추천된다

현재 라이브 스트리밍에 사용하기에 가장 좋은 프로토콜은 HLS다