

MPEG

DASH

mpeg-DASH



Dynamic Adaptive Streaming over HTTP

목차

1) DASH 란?

2) 서비스 구조

3) MPD 란?

4) 상세 동작

5) HLS 와 비교

6) 구현 예시

1) DASH 란?



DASH는 HTTP를 사용한 적응형 동적 스트리밍이다.

사용자가 자신이 원하는 품질로

[VOD 또는 라이브 스트리밍 서비스]를

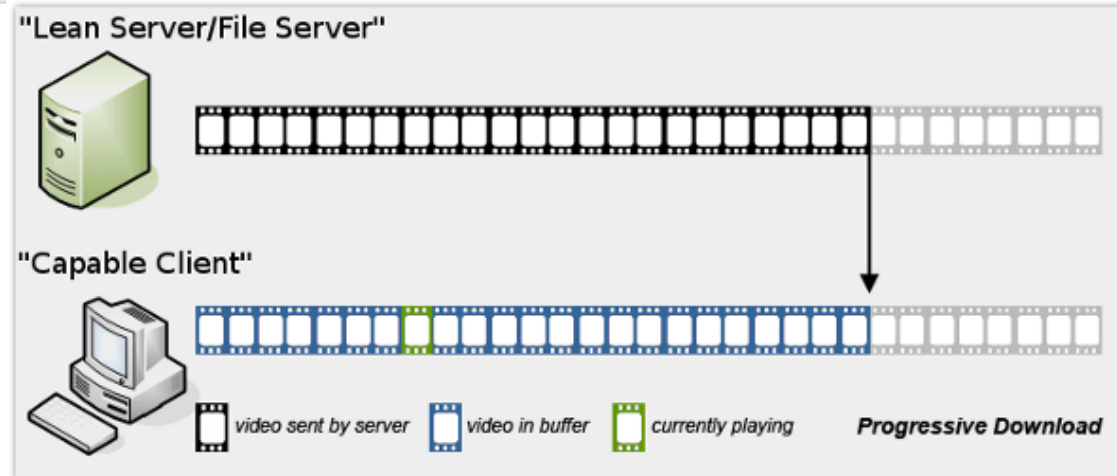
이용 받을 수 있도록 하는 프로토콜

1) DASH 란?

Adaptive HTTP Streaming 방식으로의 변화

기존 방식

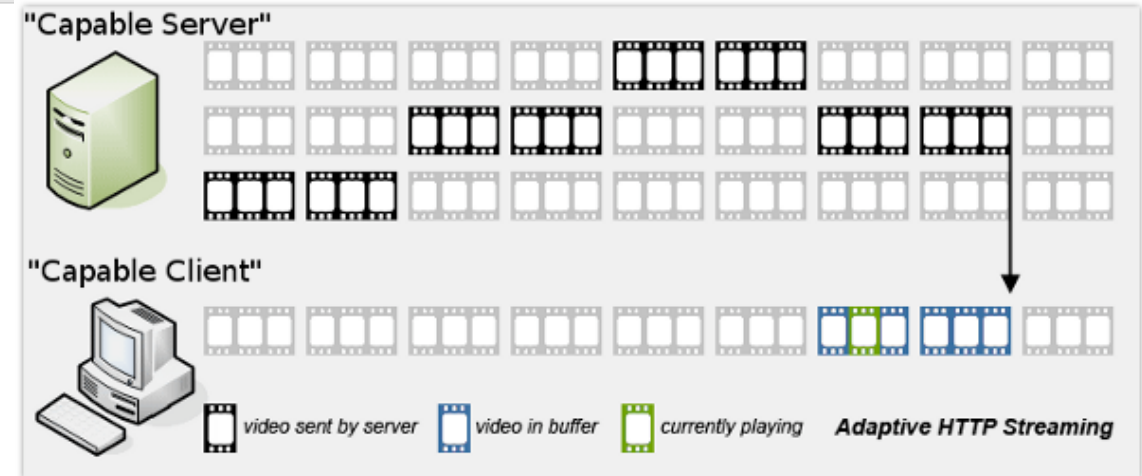
PD (Progressive Download)



1. 한번에 한가지 품질의 콘텐츠만 재생
2. 사용자가 선택한 미디어의 품질이 높을수록 네트워크 제약을 많이 받으므로, 네트워크 상황이 악화되면, 미디어 버퍼링이 불가피함
3. 콘텐츠 품질 선택권이 서버에게 있음

새로운 방식

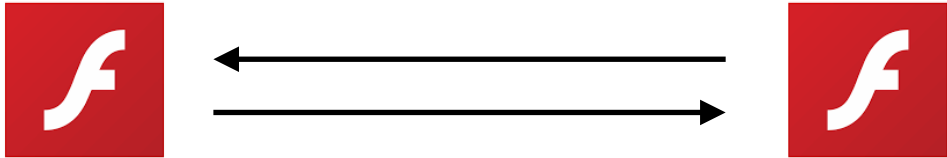

Adaptive HTTP Streaming



1. 다양한 해상도의 콘텐츠를 미리 인코딩 & 여러 개의 파일로 쪼개서 세그먼트 형태로 저장
2. 사용자에게 필요한 품질의 콘텐츠를 요청 가능
3. 콘텐츠 품질 선택권이 클라이언트에게 있음

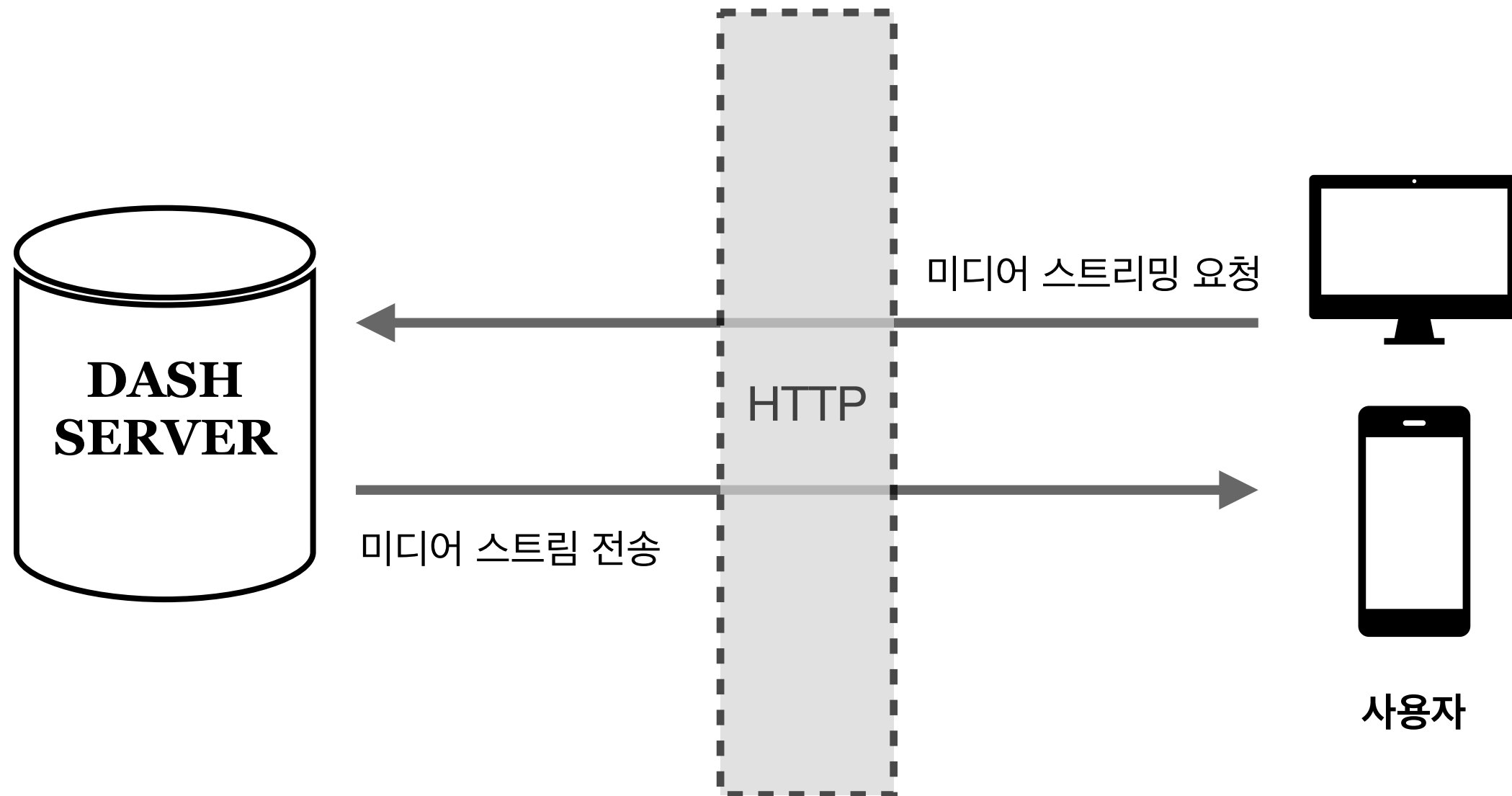
1) DASH 란?

Web 표준 방식으로 접근성 향상

기존 방식	새로운 방식
Program (예: Flash player & Silverlight)	HTTP Tunneling
	
<ol style="list-style-type: none">1. 스트리밍이 특정 프로그램에 한정적2. 네트워크 통신에 대한 접근성 낮음 > NAT 및 방화벽 환경에 대해 고려 필요	<ol style="list-style-type: none">1. HTTP 를 통해 스트리밍2. 네트워크 통신에 대한 접근성 높음 > 웹 표준 규격을 통해 통신하므로 제약이 없음

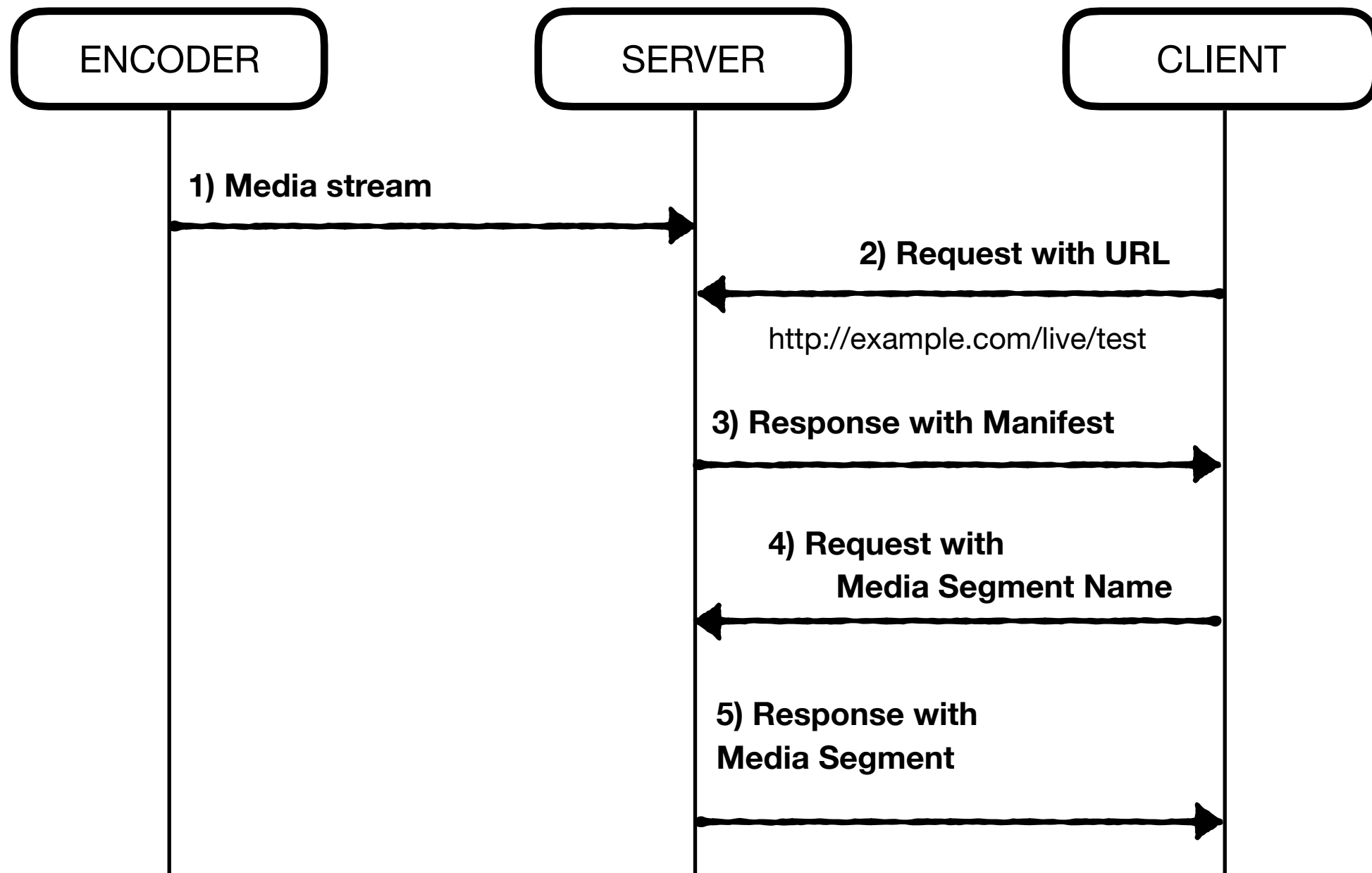
2) 서비스 구조

HTTP를 이용한 Dash Service



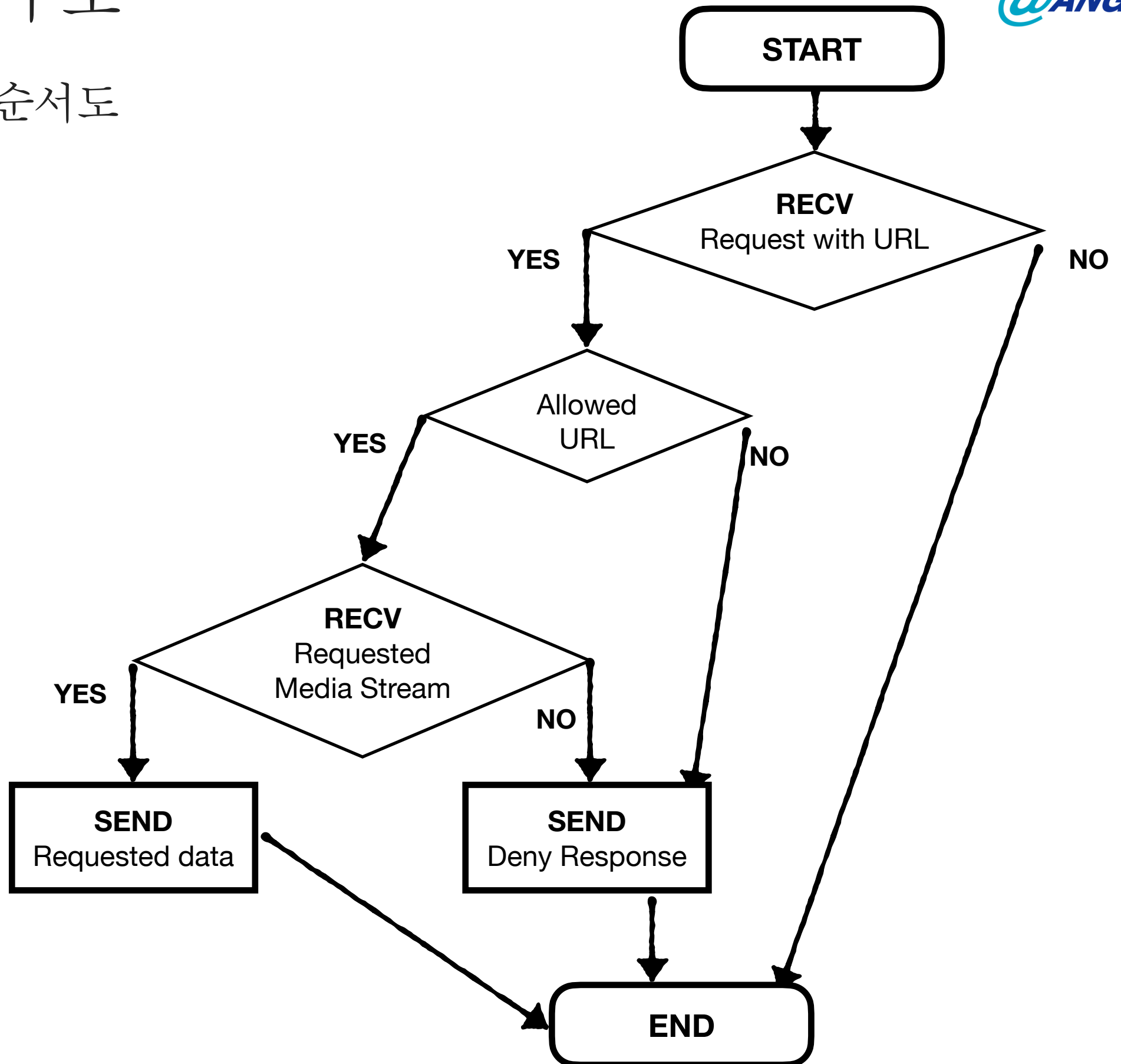
2) 서비스 구조

Dash Basic Service Flow



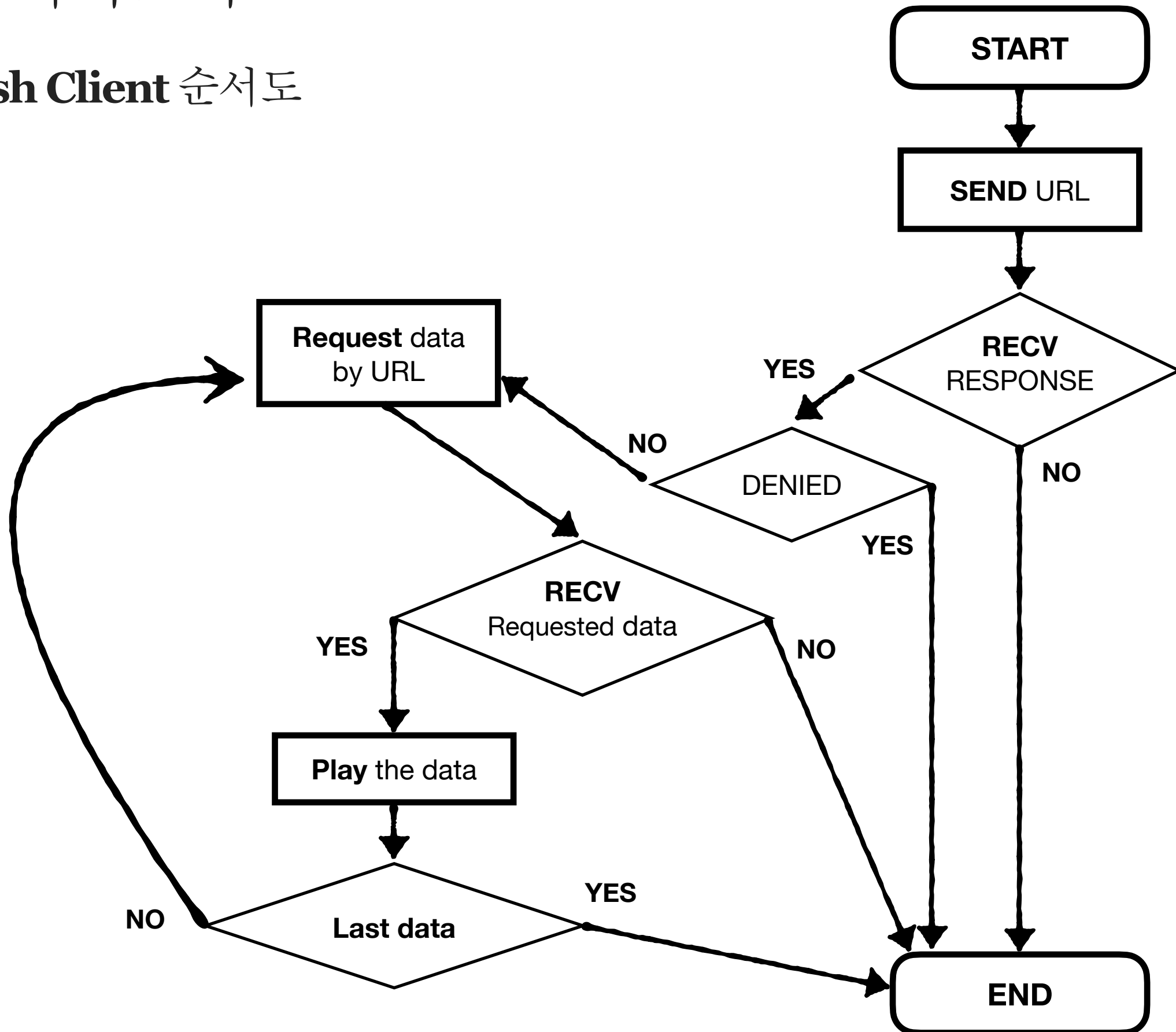
2) 서비스 구조

Dash Server 순서도



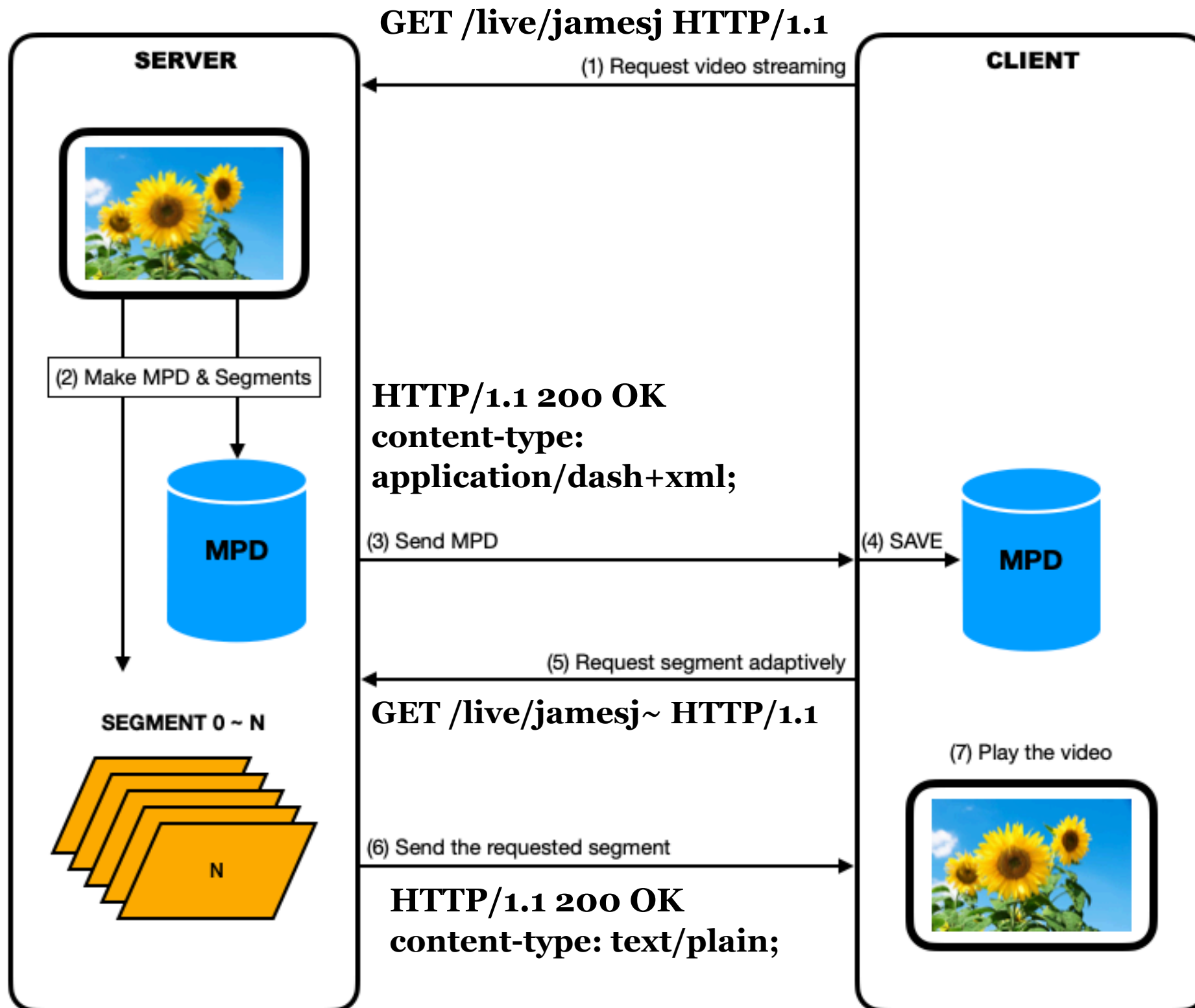
2) 서비스 구조

Dash Client 순서도



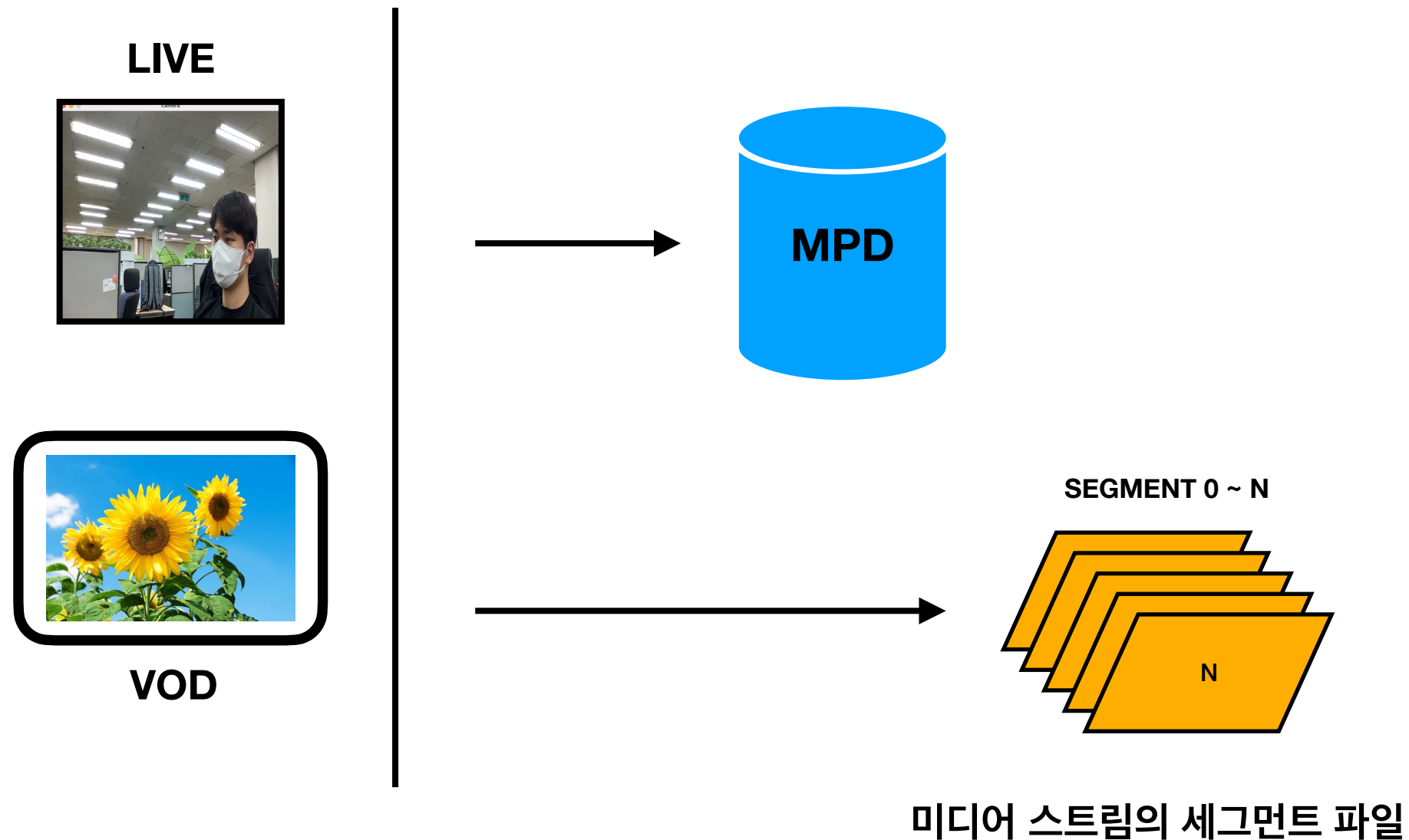
2) 서비스 구조

Dash Specific Service Flow



3) MDP 란?

MPD 는 미디어 스트림의 메타 데이터 파일이다.
(Media Presentation Description)

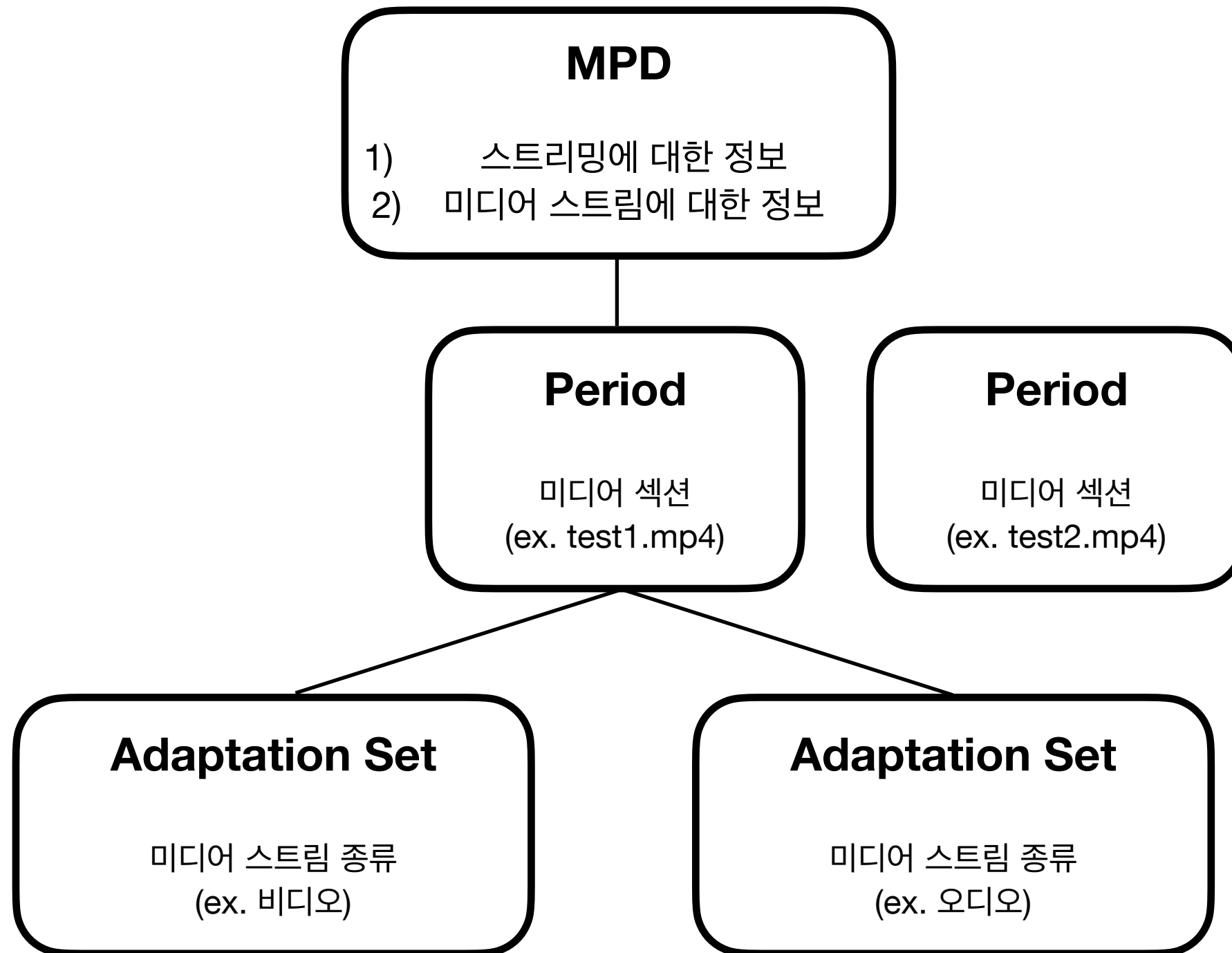


3) MDP 란?

```
<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH_schema_f
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
  type="dynamic"
  minimumUpdatePeriod="PT500S"
  availabilityStartTime="2022-03-13T23:57:24.755Z"
  publishTime="2022-03-13T23:58:07.387Z"
  timeShiftBufferDepth="PT19.9S"
  maxSegmentDuration="PT4.0S"
  minBufferTime="PT0.0S">
  <ProgramInformation>
  </ProgramInformation>
  <ServiceDescription id="0">
    <Latency target="3000"/>
  </ServiceDescription>
  <Period id="0" start="PT0.0S">
    <AdaptationSet id="0" contentType="audio" startWithSAP="1" segmentAlignment="true" bitstreamSwitching="true">
      <Resync dT="23220" type="0"/>
      <Representation id="0" mimeType="audio/mp4" codecs="mp4a.40.2" bandwidth="192000" audioSamplingRate="44100">
        <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
        <ProducerReferenceTime id="0" inband="true" type="captured" wallClockTime="2022-03-13T23:57:24.755Z"
          <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014" value="https://time.akamai.com/?iso" />
        </ProducerReferenceTime>
        <Resync dT="23220" type="1"/>
        <SegmentTemplate timescale="1000000" duration="4000000" availabilityTimeOffset="3.977" availabilityStart="2022-03-13T23:57:24.755Z" />
      </Representation>
    </AdaptationSet>
  </Period>
  <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014" value="https://time.akamai.com/?iso" />
</MPD>
```

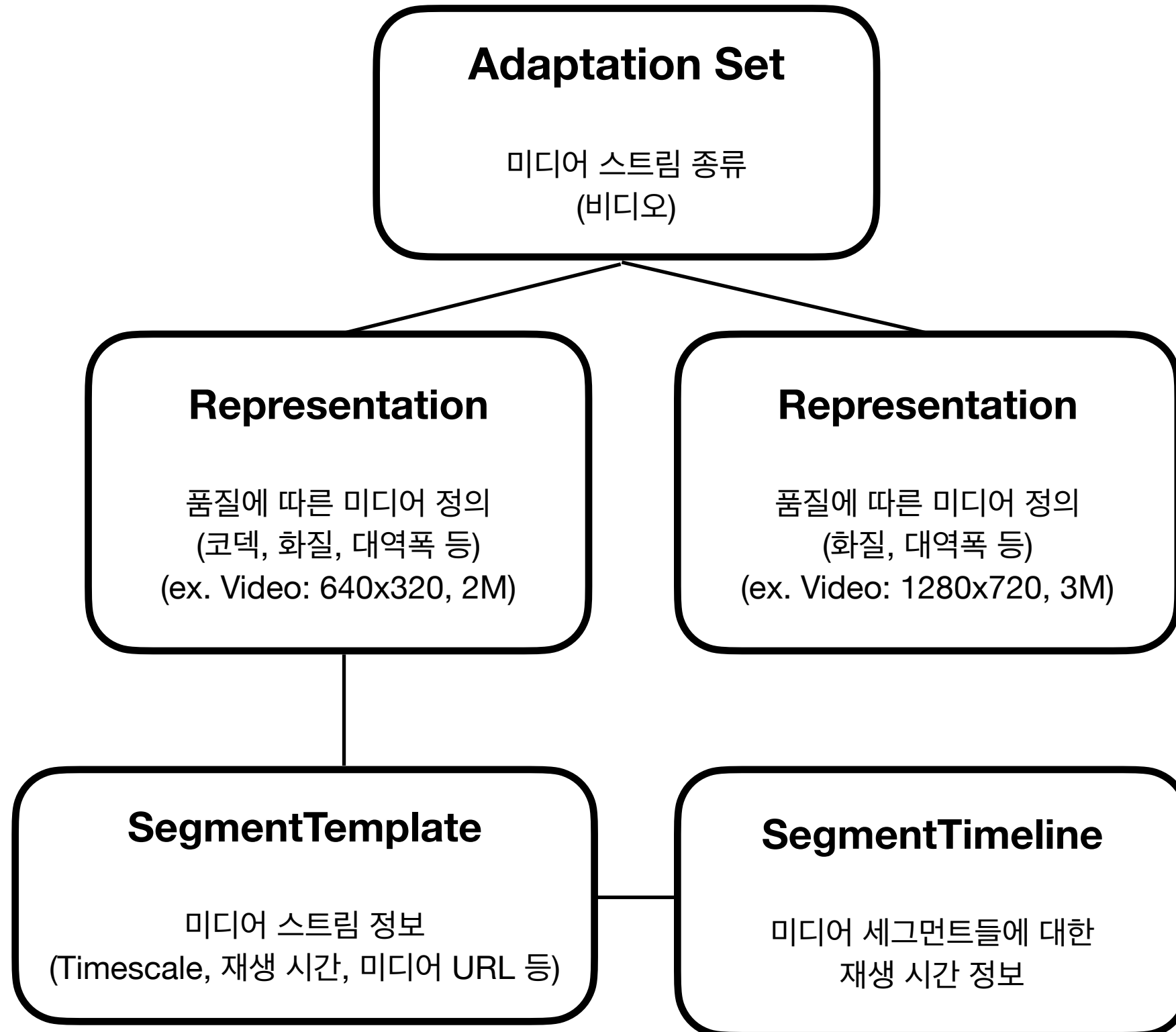
3) MDP 란?

MPD 상위 계층



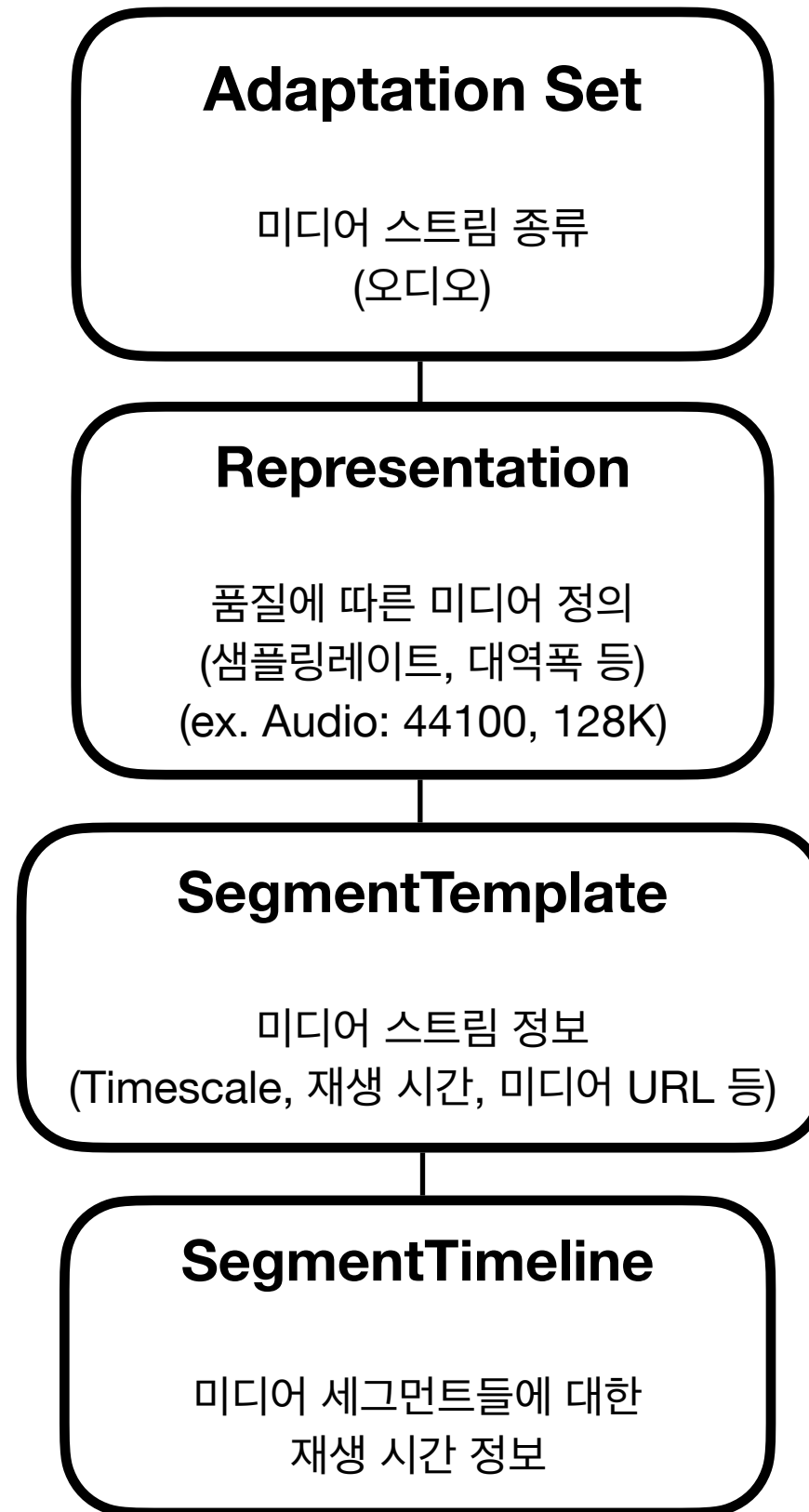
3) MDP 란?

MPD 하위 계층 : 비디오



3) MDP 란?

MPD 하위 계층 : 오디오



3) MDP 란?

MPD 상위 계층의 핵심 필드 설명

MPD	profiles	- VOD 또는 Live 스트림에 해당하는 DASH Profile 을 정의 각 미디어 유형에 따라 MPD 정의에 필요한 필드에 대한 정보를 명시 > [urn:mpeg:dash:profile:isoff-live:2011] 프로파일이 주로 사용된다.
	type	- 미디어 스트리밍 유형 > static : VOD > 세그먼트들에 대한 정보가 고정되어 있다. > dynamic : Live > 새로운 세그먼트들에 대한 정보가 계속해서 업데이트된다.
	minimumUpdatePeriod	- 클라이언트가 MPD 를 업데이트하기 위한 최소 시간 간격 을 의미
	availabilityStartTime	- Dynamic type 일 때, MPD 타임라인의 영점 을 정의한다. 인코더에서 발생한 Wall clock 에 따른 redundancy 를 최대한으로 줄이기 위해, UTC Time (영국을 기준(UTC+0:00)으로 각 지역의 시차를 규정한 시간) 을 기준 으로, 인코더와 디코더와의 시간 기준을 맞추어서 미디어 스트림을 동기화하도록 하기 위한 UTC Time 정의
	publishTime	- MPD 가 생성된 시각 을 정의. Wall clock 으로 정의 (즉, DASH 서버에서 MPD 를 생성한 후의 시각)

3) MDP 란?

MPD 하위 계층의 핵심 필드 설명

Segment Template	timescale	<ul style="list-style-type: none">- 미디어 세그먼트 재생 시간 간격을 정의한다.- 해당 시간 간격 주기를 통해 클라이언트가 세그먼트를 재생한다.
	duration	<ul style="list-style-type: none">- 미디어 세그먼트 재생 시간을 정의한다.- 클라이언트는 이 재생 시간까지 해당 미디어 세그먼트를 재생한다.
	availability Time Offset	<ul style="list-style-type: none">- 클라이언트로 하여금 미리 다음 미디어 세그먼트를 다운받을 수 있는 시간을 정의한다.- 보통 duration 보다 적은 시간으로 정의되고, preprocessing signal 역할을 한다.

3) MDP 란?

MPD 하위 계층의 핵심 필드 설명

Segment Timeline	S : 특정 미디어 세그먼트에 대한 타임라인 정의	t	<ul style="list-style-type: none">- 미디어 청크 시작 시간 (첫 번째 미디어 세그먼트의 상대적인 시작 시간)- [availabilityStartTime] 시간에서 0 μS 이후부터 재생한다는 의미
		d	<ul style="list-style-type: none">- [t] 시간에서 어느 시간(μS) 만큼 재생해야하는지에 대한 시간 정의 (미디어 세그먼트 하나당 재생 시간)
		r	<ul style="list-style-type: none">- 동일한 [d] 시간을 갖는 미디어 세그먼트들이 존재하면, MPD Redundancy 를 줄이기 위해 명시된 중복된 S 필드 카운트

3) MDP 란?

MPD 상위 계층의 핵심 필드 예시

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<MPD
```

```
...
```

```
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
```

```
  type="dynamic"
```

```
  minimumUpdatePeriod="PT500S"
```

```
  availabilityStartTime="2022-02-25T02:52:14.304Z"
```

```
  publishTime="2022-02-25T02:52:32.688Z"
```

```
...
```

3) MDP 란?

MPD 하위 계층의 핵심 필드 예시 (SegmentTemplate)

...

<Representation id="0" mimeType="video/mp4" ... >

<SegmentTemplate

timescale="1000000"

duration="5000000"

availabilityTimeOffset="4.967" ... >

</SegmentTemplate>

</Representation>

...

3) MDP 란?

MPD 하위 계층의 핵심 필드 예시 (SegmentTimeline)

...

<Representation id="0" mimeType="video/mp4" ... >

<SegmentTemplate ... >

<SegmentTimeline>

<S t="0" d="76800" r="41" />

...

<SegmentTimeline>

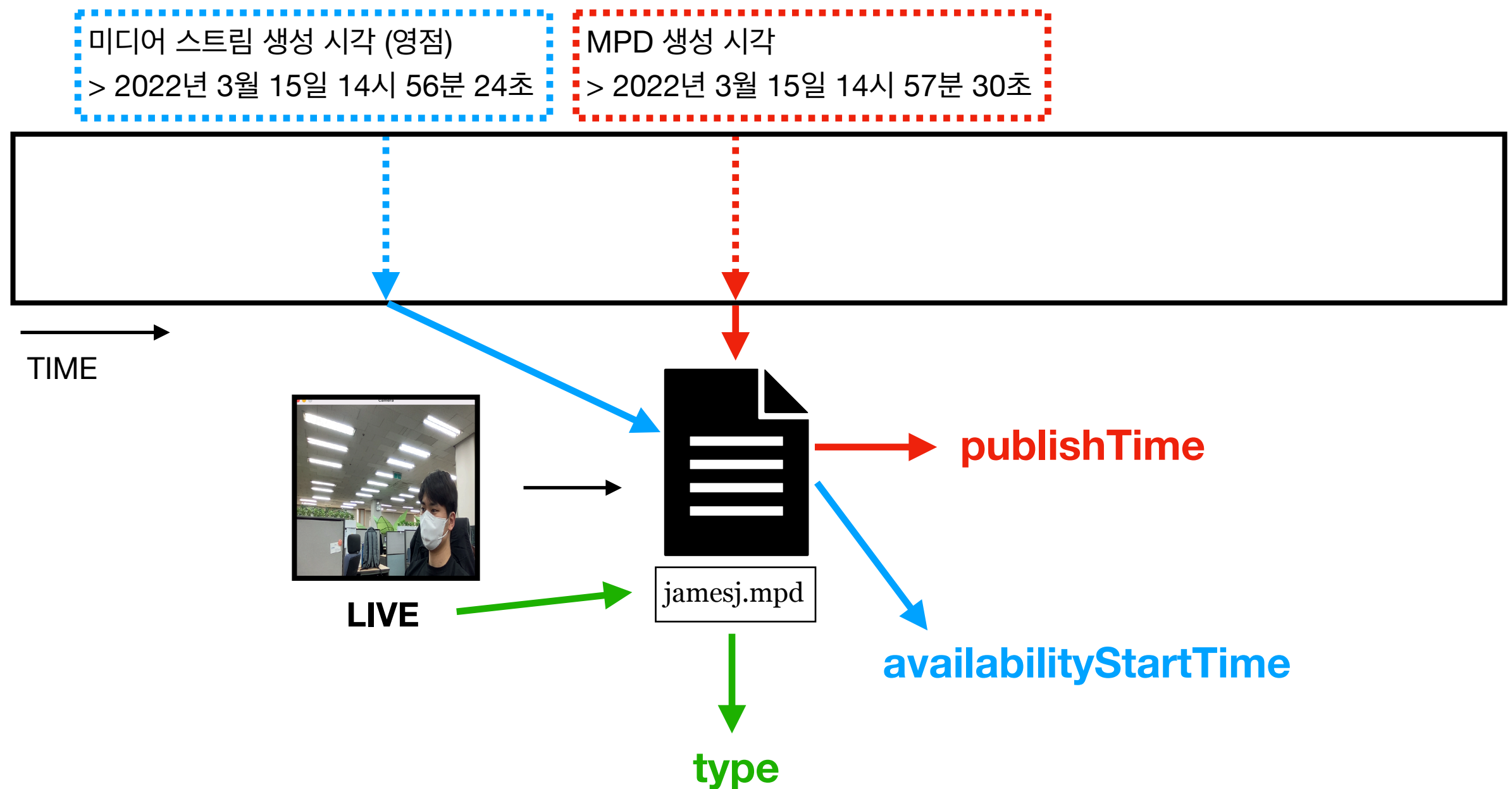
</SegmentTemplate>

</Representation>

...

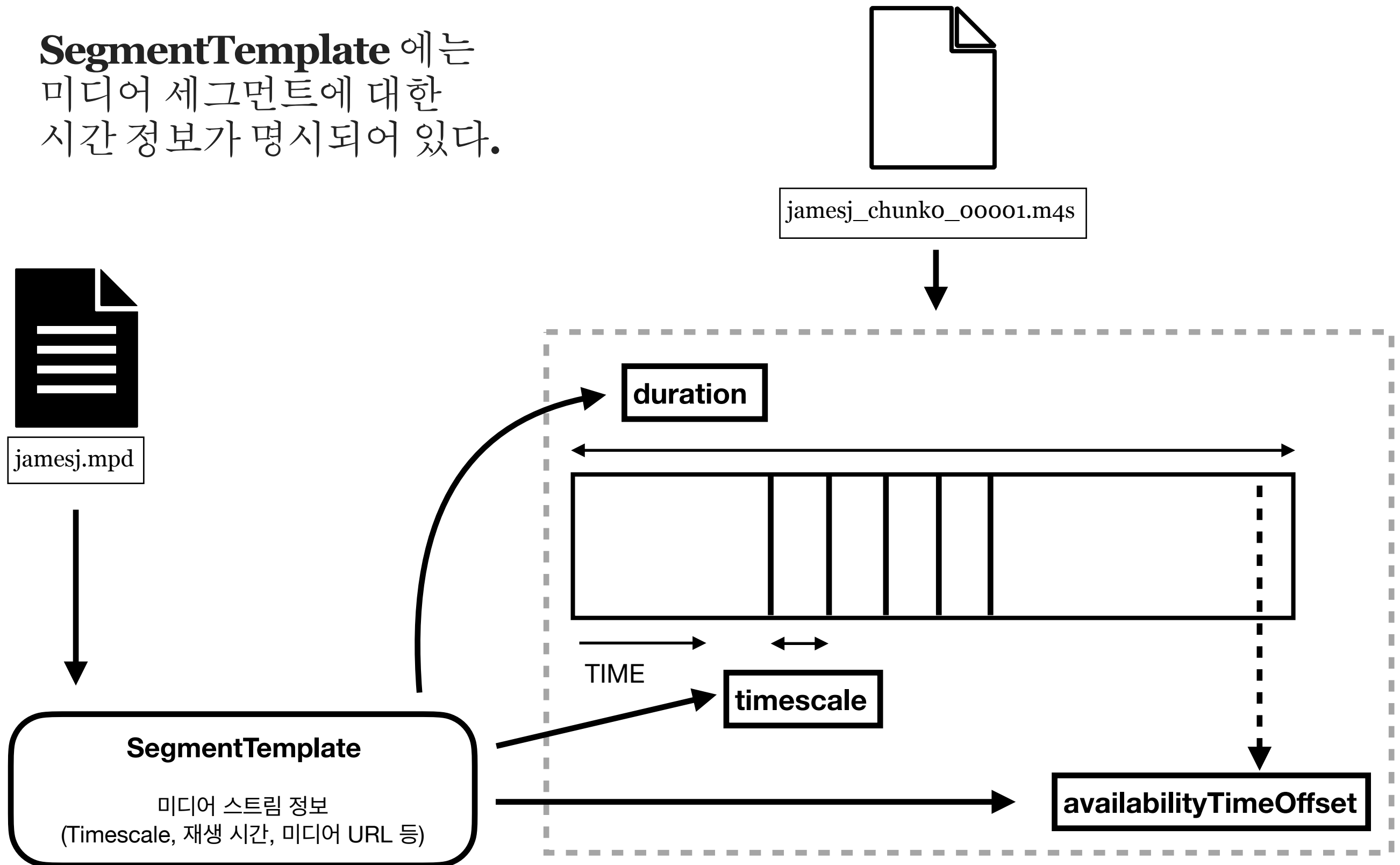
4) 상세 동작

미디어 스트림과 **MPD** 가 생성된 시각을 모두 **MPD** 에 기록한다.



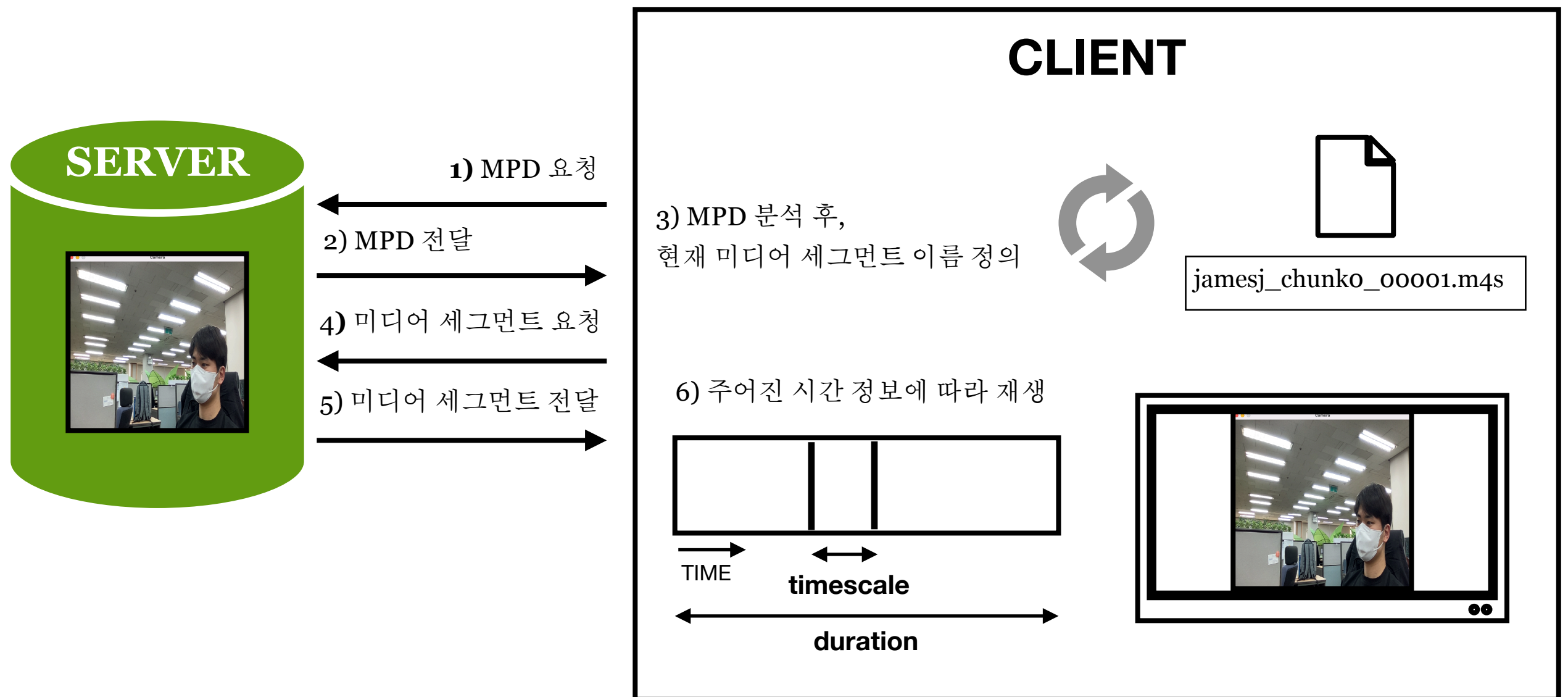
4) 상세 동작

SegmentTemplate에는
미디어 세그먼트에 대한
시간 정보가 명시되어 있다.



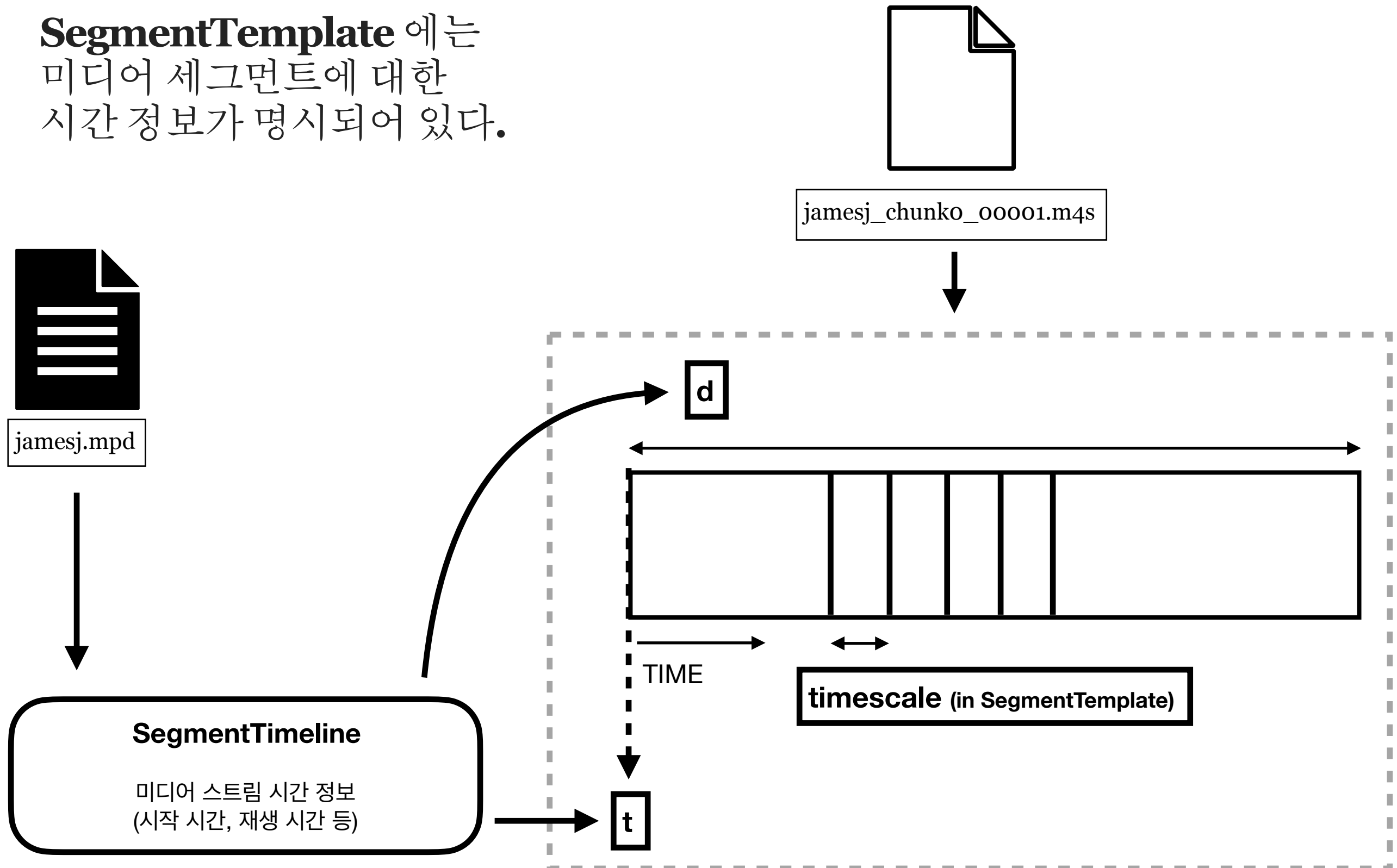
4) 상세 동작

MPD에 명시된 미디어 세그먼트 파일 정보와 시간 정보를 통해서 미디어 세그먼트를 재생한다.



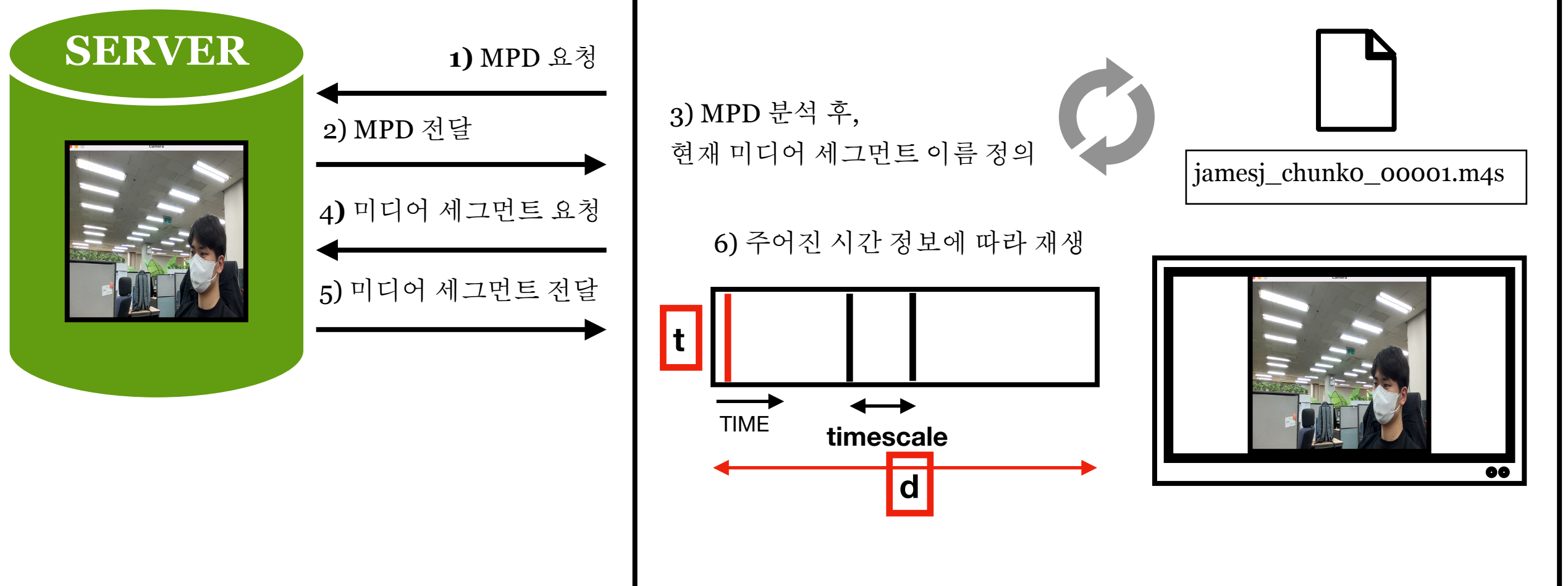
4) 상세 동작

SegmentTemplate에는
미디어 세그먼트에 대한
시간 정보가 명시되어 있다.



4) 상세 동작

MPD에 명시된 미디어 세그먼트 파일 정보와 시간 정보를 통해서 미디어 세그먼트를 재생한다.

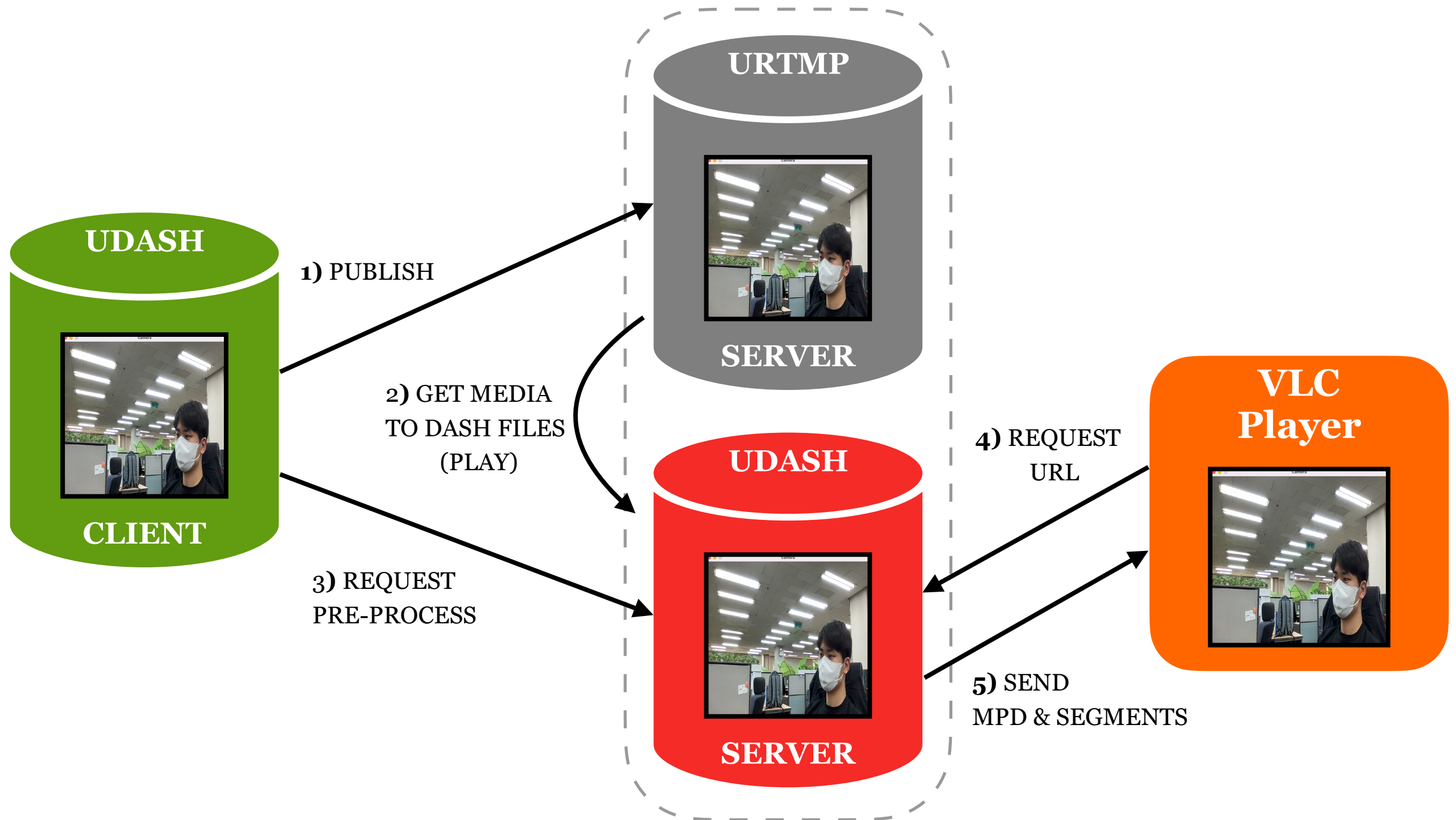


5) HLS 와 비교

구분	HLS	DASH
제조사	Apple	MPEG
호환성	제약 없음	MSE 사용 플랫폼만 사용 가능 (IOS 는 MSE 지원 안함)
통신 프로토콜	HTTP	HTTP
지원 코덱	비디오 : H.264, H.265 오디오 : AAC, Dolby	제약 없음
암호화	AES-128 (https://www.dacast.com/blog/hls-encryption-for-video/)	CENC (https://docs.unified-streaming.com/documentation/drm/common-encryption.html#common-encryption)
고도화	Low Latency HLS (LL-HLS) with CMAF (Comon Media Application Format)	Low Latency DASH (LL-DASH) with CMAF

6) 구현 예시

[UDASH] 와 [URTMP_SERVER] 를 이용한 시뮬레이션



감사합니다.

[REFERENCE]

1. HLS vs DASH > <https://meetup.toast.com/posts/131>
2. HLS > <https://yoooonghyun.gitbook.io/documents/multimedia/overview/hls>
3. HLS > <https://datatracker.ietf.org/doc/html/rfc8216>
4. RTMP > <https://cdmana.com/2021/04/20210429181340831F.html>
5. RTMP > <https://drive.google.com/file/d/oBx-jA7MdEVHIZ3dmb1RoR3JkUTg/view?resourcekey=o-Cz7EXmSVJo1NR-a2893YJA>
6. DASH > <https://dgo87.tistory.com/85>
7. DASH > <https://mpeg.chiariglione.org/standards/mpeg-dash>
8. DASH > <https://bitmovin.com/dynamic-adaptive-streaming-http-mpeg-dash/>
9. MSE > <https://yethor.tistory.com/11>