



# THEO WEBINAR

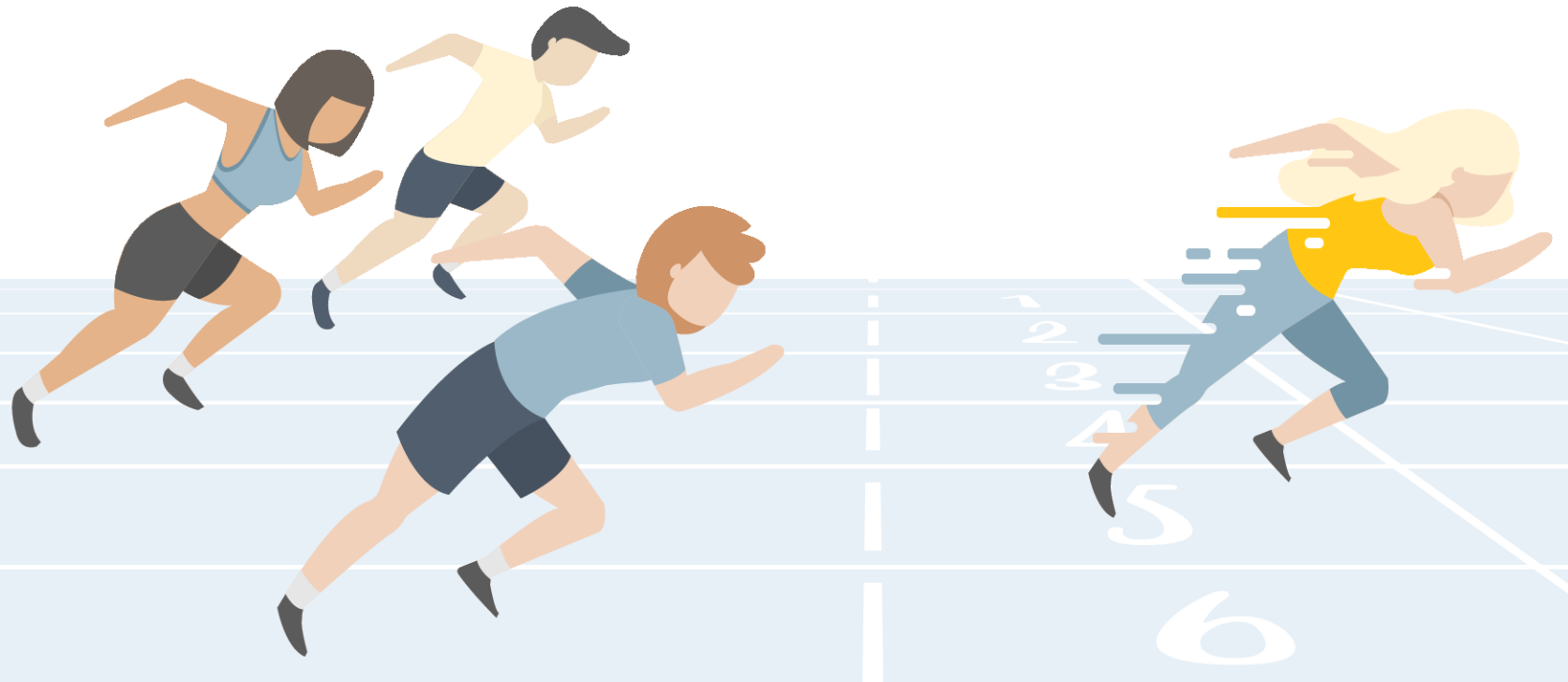
## LOW LATENCY CHECK-UP



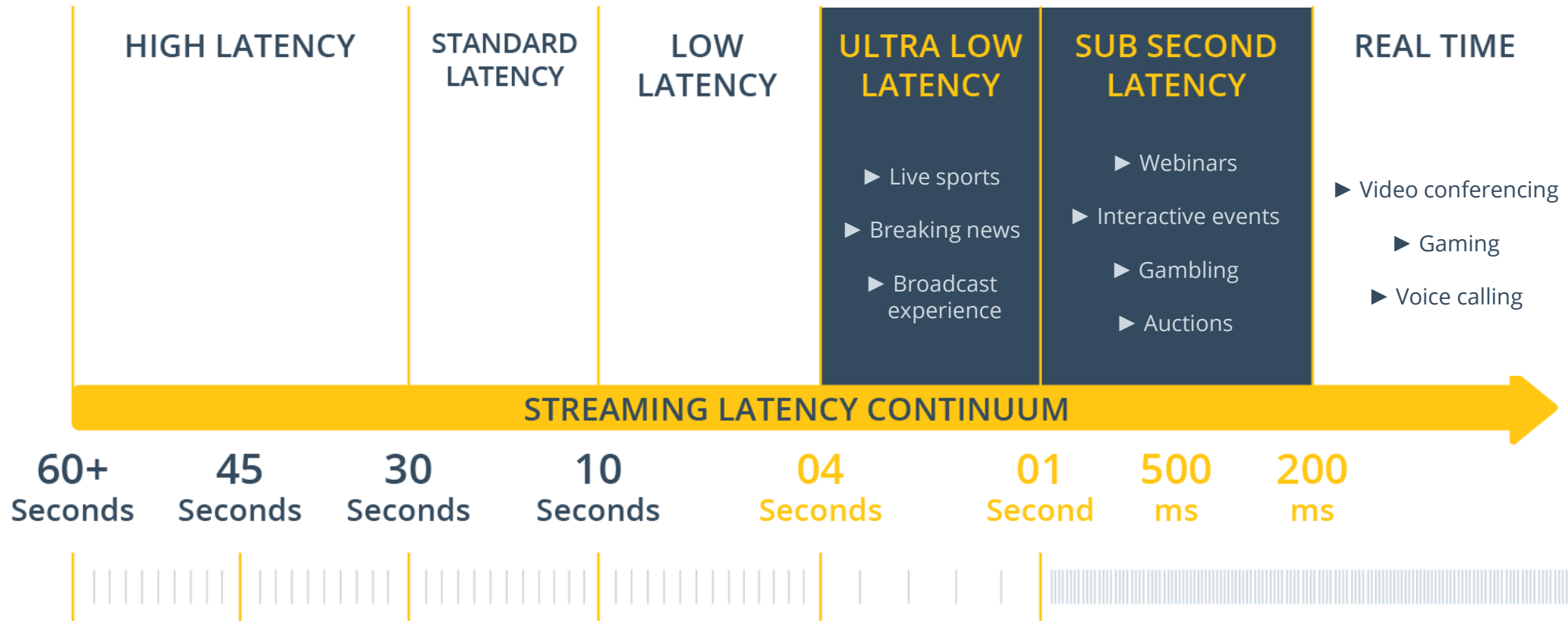
PIETER-JAN SPEELMANS  
CTO & Founder



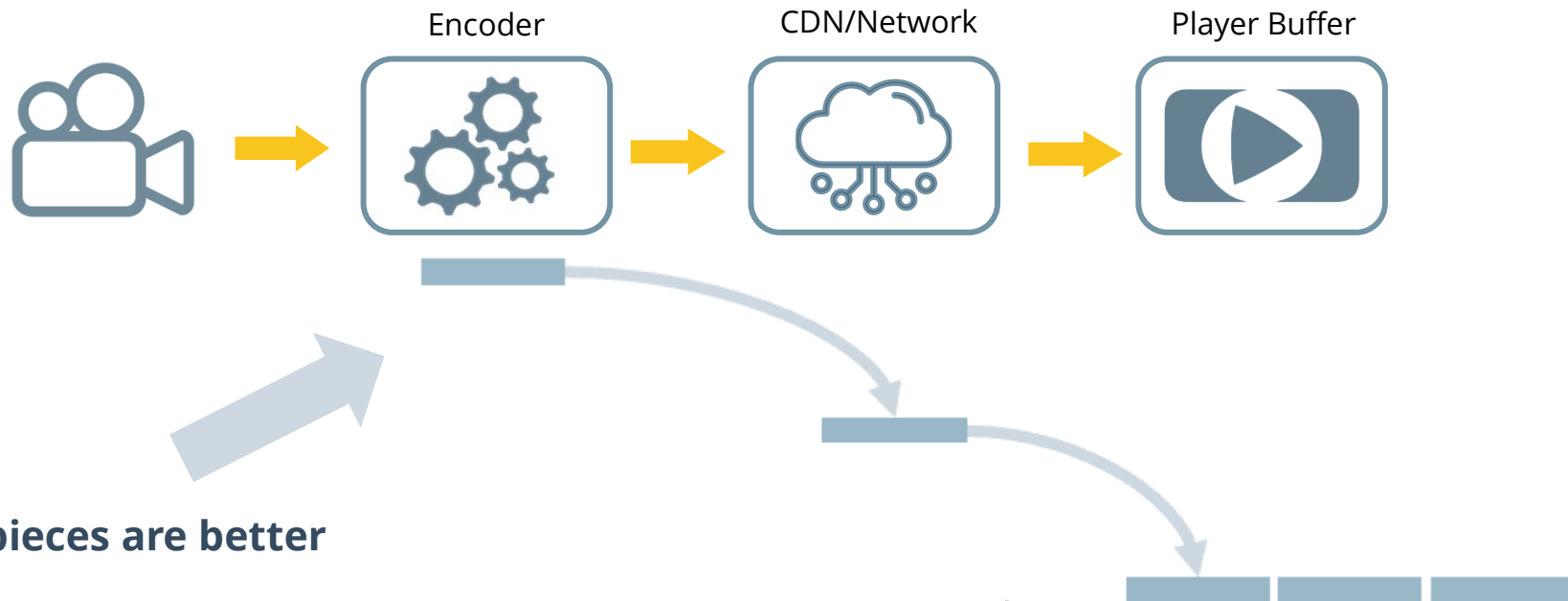
JOHAN VOUNCKX  
VP Innovation



# LATENCY, MOSTLY DUE TO THE PROTOCOL

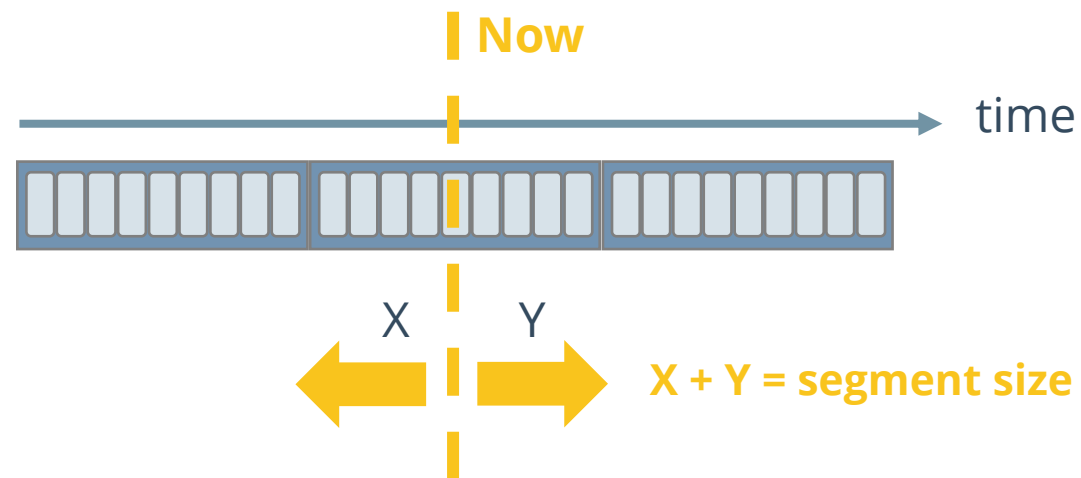


# THE ORIGINS OF LATENCY



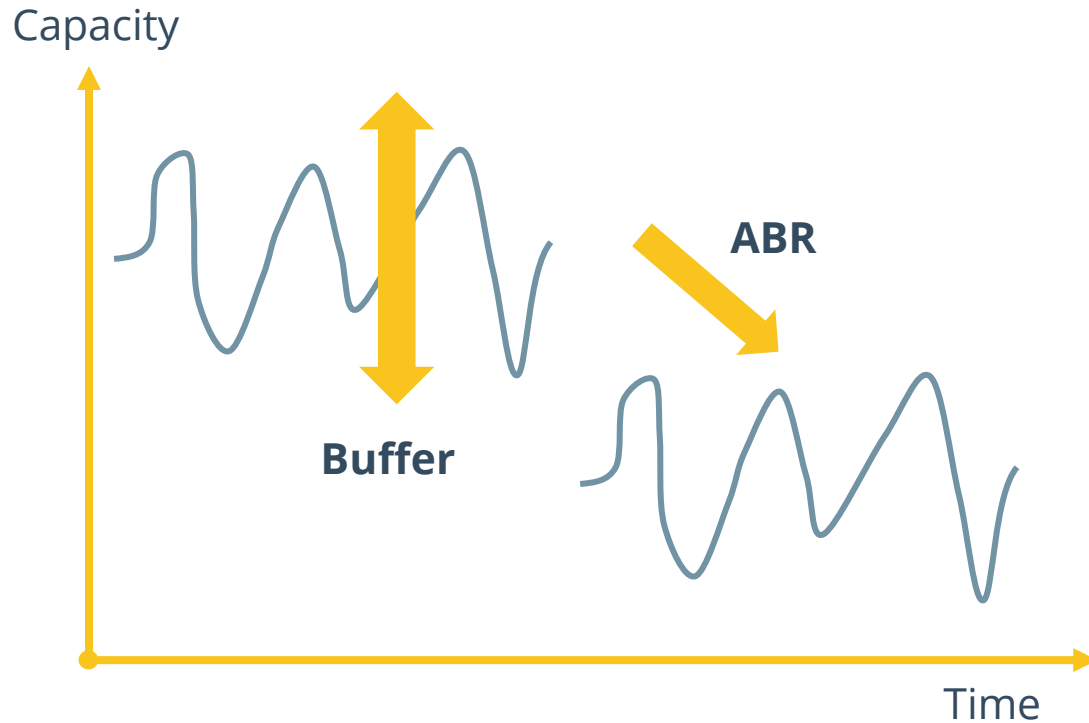
# LATENCY IS NOT THE ONLY CRITERION - START-UP TIME

- ▶ Need an IDR to start (typically beginning of a segment)
- ▶ Leads to a trade-off between startup latency and startup time.
- ▶ More IDRs make it easier to combine low startup latency and startup time
- ▶ But ... more IDRs (shorter GOPs) imply higher bitrates / lower quality for the same bitrate.



Either X live latency (possibly at line-speed) or Y startup latency or a combination

# LATENCY IS NOT THE ONLY CRITERION - NETWORK RESILIENCE



- ▶ Network capacity changes call for ABR solutions
- ▶ ABR solutions select a more adequate quality/bitrate.
- ▶ Switchover needs an IDR.
- ▶ Buffer need to be large enough to wait for an IDR.
- ▶ High frequency network fluctuations also require a buffer (with a direct negative impact on latency)

# LATENCY IS NOT THE ONLY CRITERION - QUALITY & BITRATE

▶ Bitrate in function of the GOP size

GOP/Bitrate	0.5 sec	1 sec	2 sec	3 sec	6 sec	10 sec
Tears of Steel	134%	114%	106%	104%	100%	100%
Bike race	126%	111%	104%	102%	100%	99%
static video (webinars, ...)	427%	256%	171%	144%	111%	100%

▶ VMAF score for a constant bitrate in function of the GOP size

GOP/VMAF	0.5 sec	1 sec	2 sec	3 sec	6 sec	10 sec
Tears of Steel	79.8	83.1	84.6	85.1	85.6	85.8
Bike race	59.6	64.6	67.3	67.8	68.4	68.
static video (webinars, ...)	95.8	95.8	95.8	95.9	95.9	95.9

## QUICK REFRESHER ON LL-DASH

- ▶ LL-DASH spec available since 2018
- ▶ Published by the DASH Industry Forum (<https://dashif.org/>)
- ▶ Segments are
  - ▶ split into fMP4 “chunks” and
  - ▶ delivered over HTTP CTE (chunked transfer encoding)
  - ▶ anticipated using availability start times
- ▶ Playback can start by selecting a segment and loading one after the other
  - ▶ Latency at start-up is mostly influenced by segment size
  - ▶ Channel switching is influenced by segment size

## QUICK REFRESHER ON LL-HLS

- ▶ LL-HLS spec available since 2020 (<https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis-09>)
- ▶ Published by Apple
- ▶ Segments are
  - ▶ Split into fMP4 or TS “parts” and
  - ▶ Delivered over HTTP/2 at line speed
  - ▶ Anticipated using blocking requests and EXT-X-PRELOAD-HINT
- ▶ Playback can start by selecting an independent part and loading one after the other
  - ▶ Latency at start-up is mostly influenced by GOP size
  - ▶ Channel switching is influenced by GOP size



## QUICK REFRESHER ON HESP

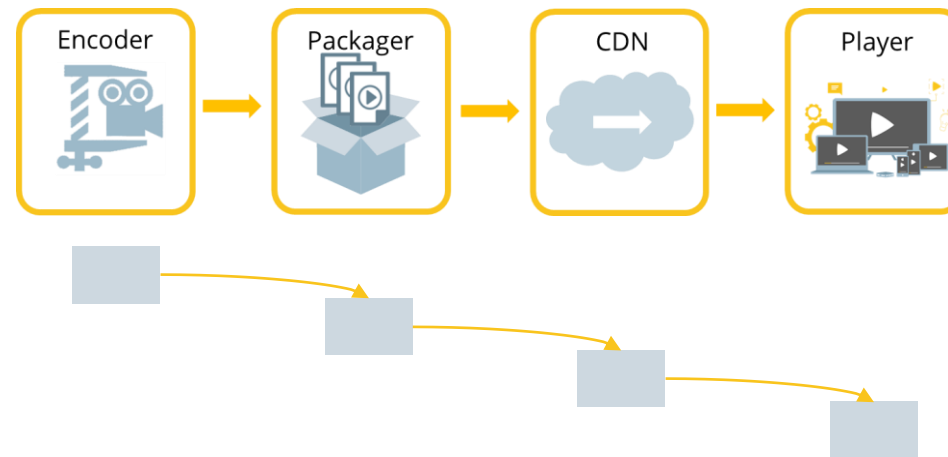
- ▶ HESP spec available since 2021 (<https://www.ietf.org/id/draft-theo-hesp-00.html>)
- ▶ Published by HESP Alliance (<https://www.hespalliance.org/>)
- ▶ Two streams are available
  - ▶ Initialization stream for fast stream switching
  - ▶ Continuation stream for highly scalable regular viewing
- ▶ Segments are
  - ▶ split into fMP4 packets and
  - ▶ delivered over HTTP CTE (chunked transfer encoding)
- ▶ Playback starts by loading a single initialization packet and issuing a range request for the continuation stream
  - ▶ Latency at start-up is always low
  - ▶ Channel switching is always extremely fast

# THE COMPLETE VIEWER EXPERIENCE MUST BE OPTIMIZED

## Ultra-Low Latency



*ULL by tuning in the video stream at the live edge and by making the images available to the player as soon as they are generated.*



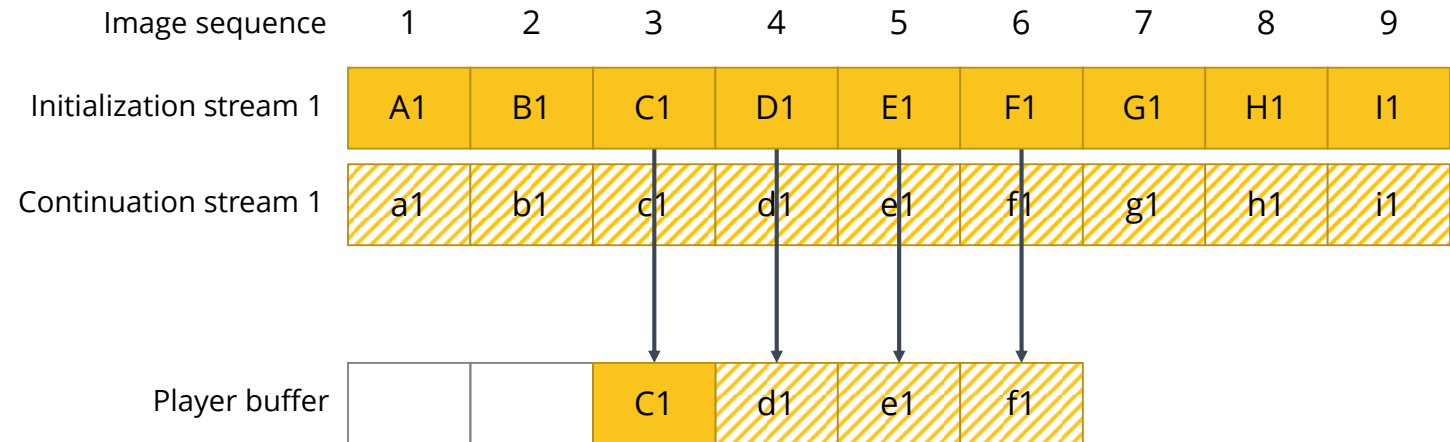
# THE COMPLETE VIEWER EXPERIENCE MUST BE OPTIMIZED

Ultra short start  
time, seeking time,  
...

...

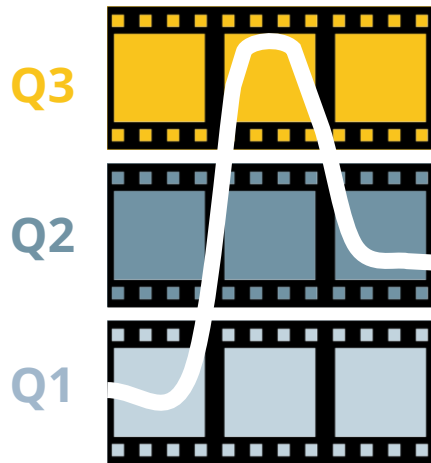


*Start at any given moment in the video, without being constrained by GOP sizes, segment sizes, ...*

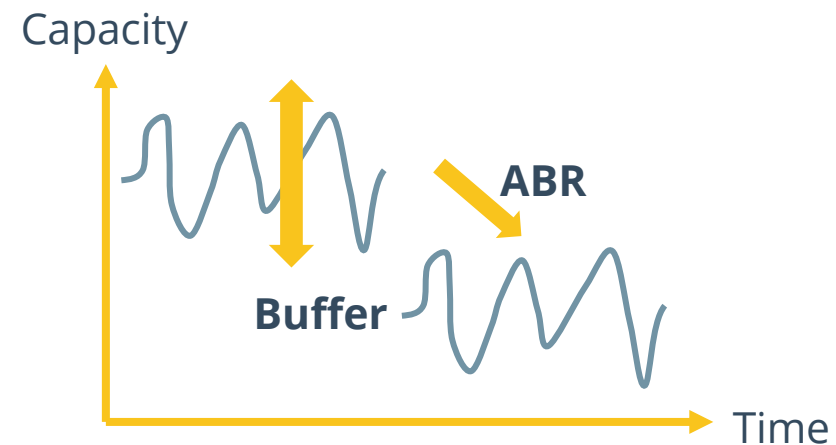


# THE COMPLETE VIEWER EXPERIENCE MUST BE OPTIMIZED

## Advanced ABR



*Instant quality switching without any stalling, in turn again allowing to reduce the buffer size.*



# THE COMPLETE VIEWER EXPERIENCE MUST BE OPTIMIZED

## Video Quality



**#1 QoE**


*Avoid impacting compression to quality ratios and reducing the GOP size while ensuring resilience to network issues. Allow the cost of distribution to scale without impacting the QoE.*

GOP/Bitrate	0.5 sec	1 sec	2 sec	3 sec	6 sec	10 sec
Tears of Steel	134%	114%	106%	104%	100%	100%
Bike race	126%	111%	104%	102%	100%	99%
static video (webinars, ...)	427%	256%	171%	144%	111%	100%

GOP/VMAF	0.5 sec	1 sec	2 sec	3 sec	6 sec	10 sec
Tears of Steel	79.8	83.1	84.6	85.1	85.6	85.8
Bike race	59.6	64.6	67.3	67.8	68.4	68.
static video (webinars, ...)	95.8	95.8	95.8	95.9	95.9	95.9


# WHAT WE NEED

**Ultra Low Latency**



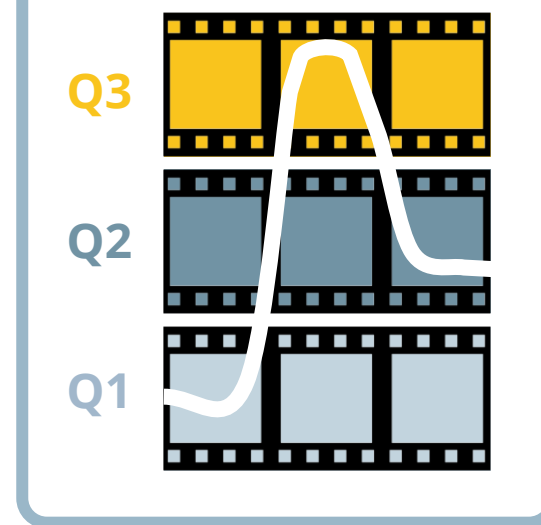
An illustration showing two hands holding a smartphone. The screen displays a tennis player in mid-swing on a tennis court. The background is a light blue gradient.

**Ultra short start time, seeking time, ...**



An illustration of a person sitting on a blue sofa in a living room, watching a television. A coffee table and a lamp are also visible. The background is a light blue gradient.

**Advanced ABR**



A diagram illustrating Adaptive Bit Rate (ABR). It shows three horizontal film strips representing different quality levels: Q3 (top, yellow), Q2 (middle, blue), and Q1 (bottom, grey). A white line graph curves across the strips, starting at Q1, rising to Q3, and then falling back to Q1, representing the dynamic adjustment of video quality based on network conditions.

**Video Quality**



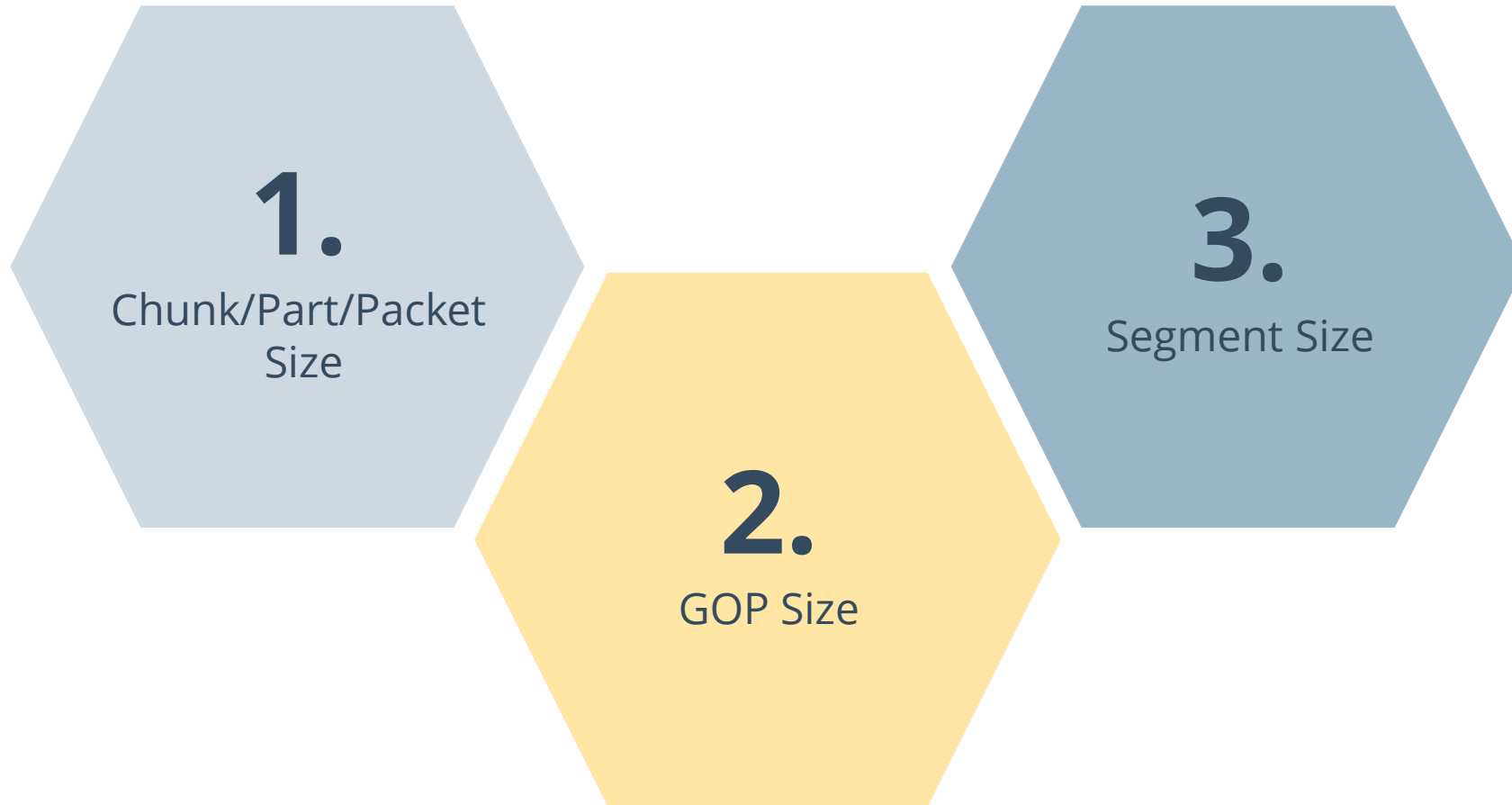
**#1 QoE**

An icon featuring a blue star in the center, surrounded by a laurel wreath, symbolizing top performance or quality.

**Scalable over standard HTTP infrastructures**

# GETTING THE MOST OF OUT YOUR PROTOCOLS

## 3 CRUCIAL PARAMETERS



# GETTING THE MOST OF OUT YOUR PROTOCOLS

## 3 CRUCIAL PARAMETERS - AS THEY IMPACT YOUR BUFFER SIZE

### Playing in a steady state

- ▶ Your buffer is your main source of latency
- ▶ You must be able to download a new unit of data:

***Buffer  $\geq$  maximal time to receive a chunk***

- ▶ You have some jitter on your network, so you need a safety margin

***Buffer = {few chunks/parts/packets}***

- ▶ Your latency can be as low as 3-4x your chunk/part/packet size

***Latency  $\geq$  3 \* chunk size***

### Playing in a variable environment

- ▶ Your buffer is needed to give the player time to switch qualities
- ▶ You must be able to download a new unit of data starting with an independent frame

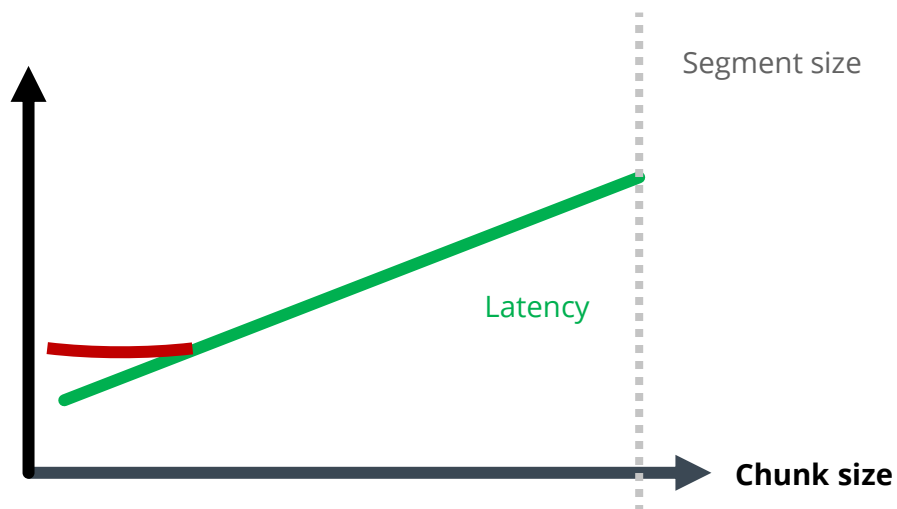
***Buffer  $\geq$  GOP size***

- ▶ You must also be able to retrieve all needed data
  - ▶ LL-DASH: Initializer
  - ▶ LL-HLS: Playlist / EXT-X-MAP
  - ▶ HESP: Initialization packet
- ▶ A safe latency is higher than the time needed to get an independent frame and the amount of RTTs you need to load all needed data



# GETTING THE MOST OUT OF LL-DASH

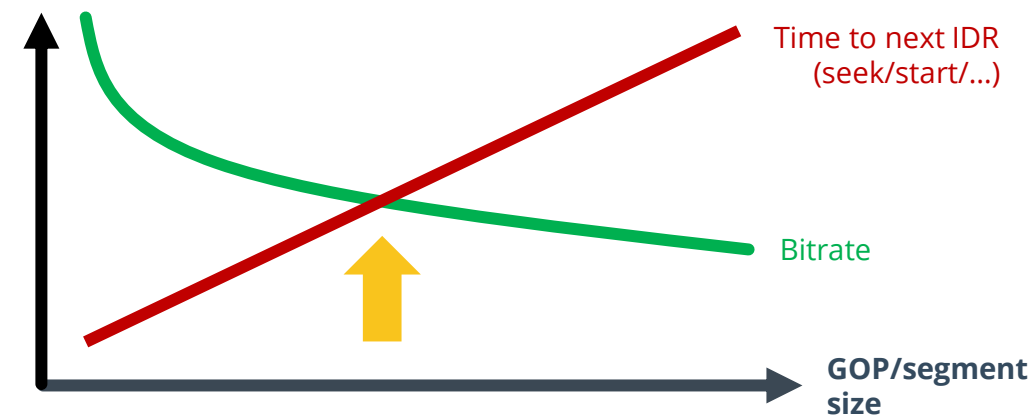
- ▶ Reducing the chunk size has its limits



- ▶ Reducing chunk below a certain point has no further impact: RTT, network variations, overhead, ...

- ▶ Recommended chunk size: 100-200ms
- ▶ Recommended GOP size: 2s
- ▶ Recommended Segment size: 2s

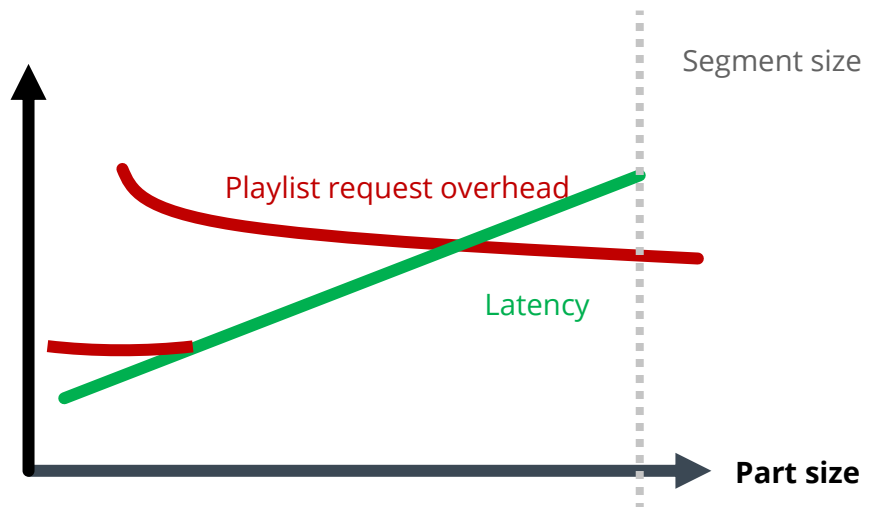
- ▶ GOP size should not be too low, and is linked to segment size



- ▶ Segment size should be the same as GOP size

# GETTING THE MOST OUT OF LL-HLS

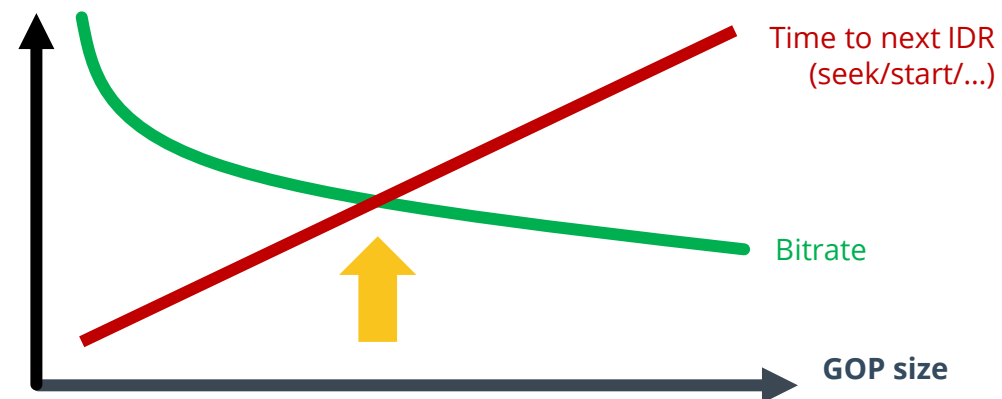
- ▶ Reducing the part size increases playlist overhead



- ▶ Overhead can become too large if the part becomes too small

- ▶ Recommended part size: 1s
- ▶ Recommended GOP size: 2s
- ▶ Recommended Segment size: 6s

- ▶ LL-HLS can decouple impact from segment size on start-up time

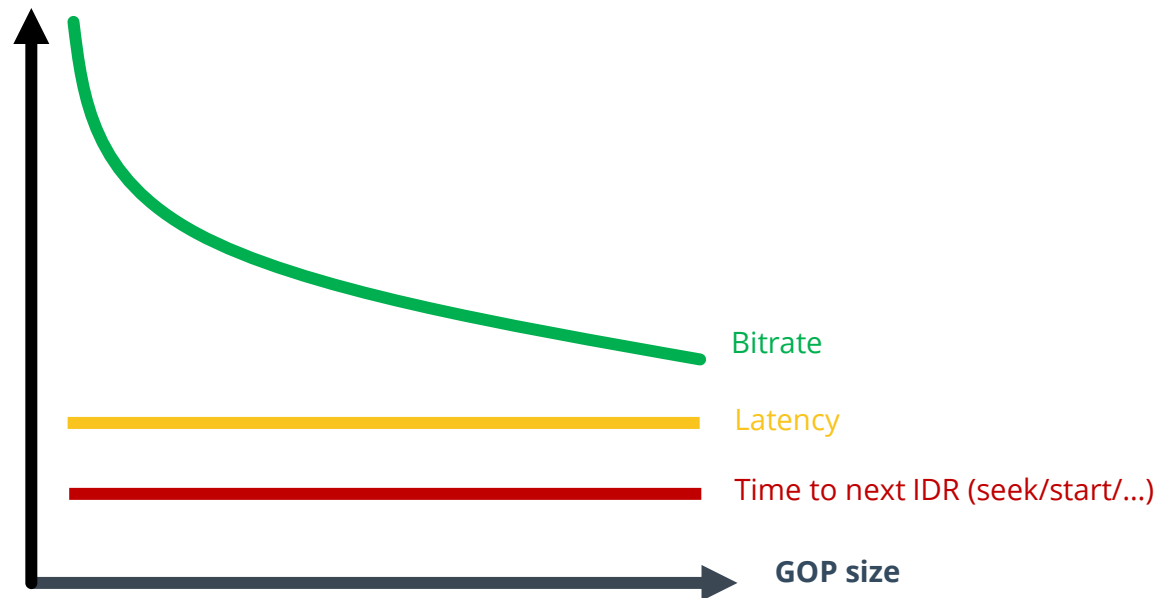


- ▶ Segment size can be larger, but value is limited (smaller playlists)

# GETTING THE MOST OUT OF HESP

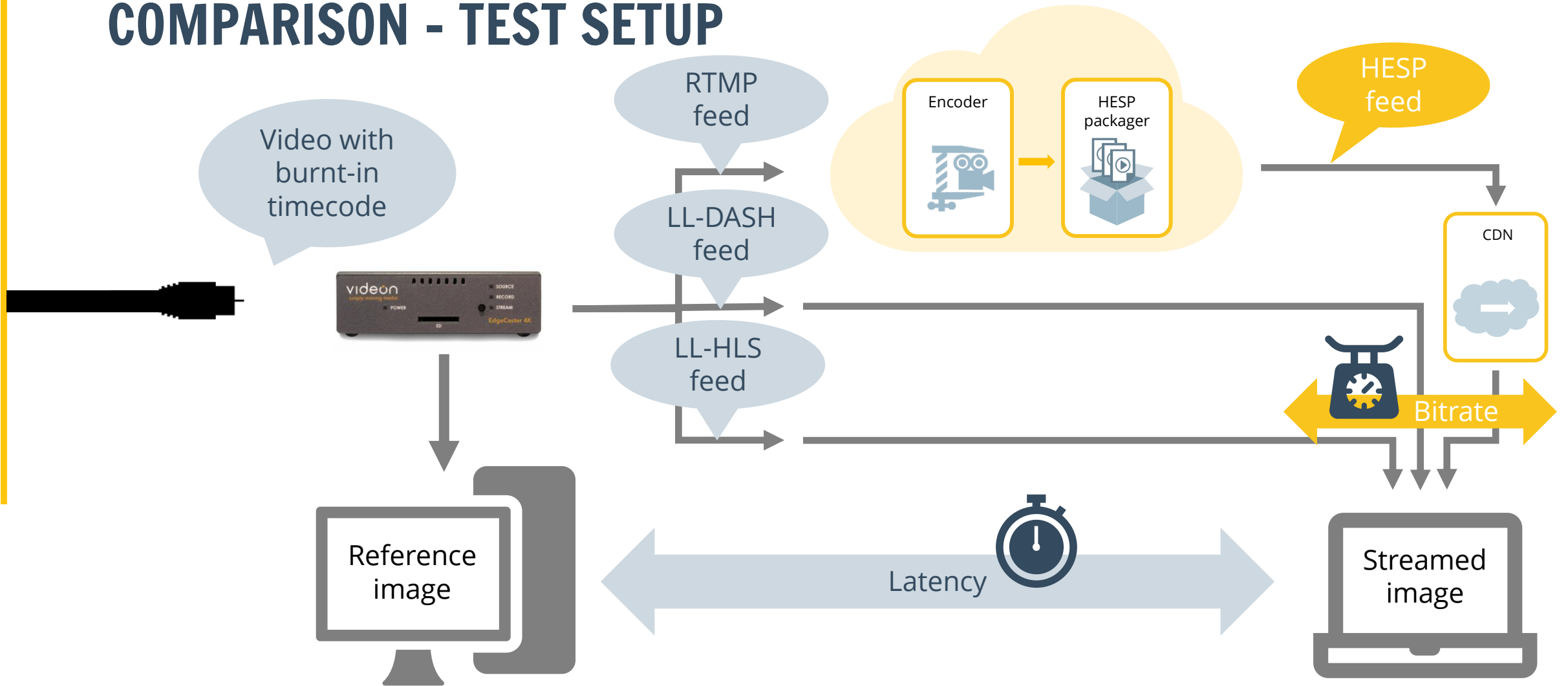
▶ Initialization and continuation feeds decouple GOP size from latency, but related mostly to RTT and the size of an initialization packet

▶ Availability of Initialization feed means buffer (and latency) can stay constant regardless of GOP size



- ▶ Recommended chunk size: 1 frame
- ▶ Recommended GOP size: 2-6s
- ▶ Recommended Segment size:  $\geq$  GOP size

# COMPARISON - TEST SETUP





# COMPARISON - TEST RESULTS

	LL-DASH	LL-HLS	HESP
GOP size	2 s	2 s	6 s
Chunk/part size	200 ms	1 s	1 frame
Segment size	2 s	6 s	24 s
Protocol latency	2.5 s	3.0 s	0.5 s
Startup time	0.5-2 s*	0.7-2 s*	0.4 s
Bandwidth	105%*	105%*	100%
#requests	60 pmin	120 pmin	3 pmin

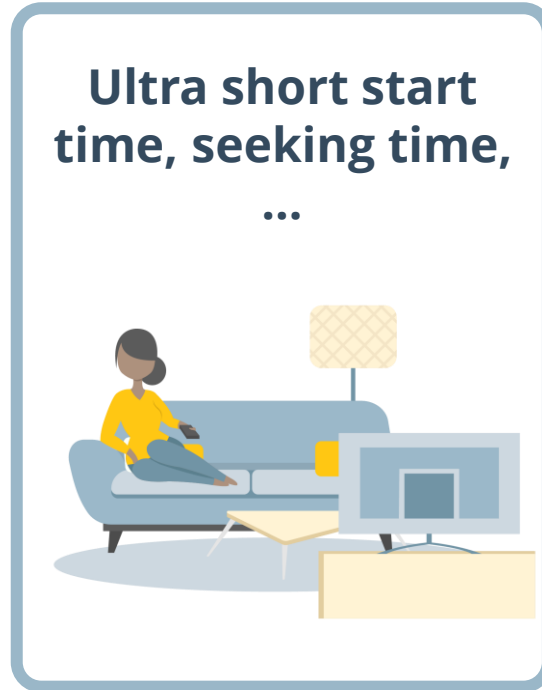
# IT'S HIGH EFFICIENCY STREAMING

**Ultra Low Latency**



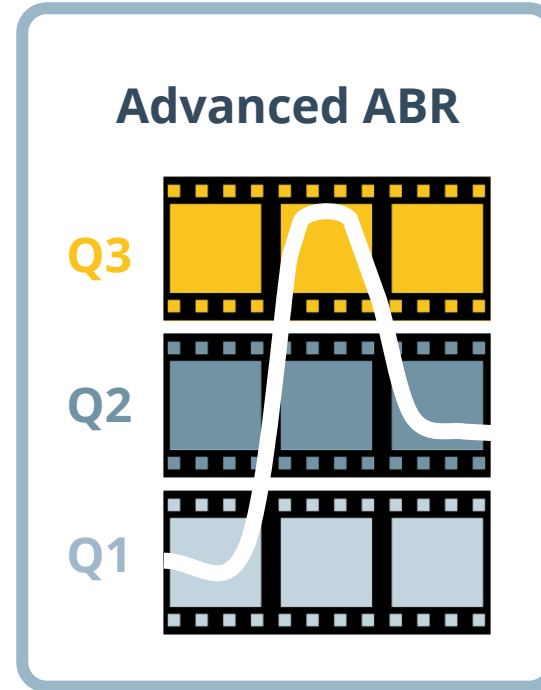
An illustration showing two hands holding a smartphone. The screen displays a tennis player in mid-swing on a tennis court. The background is a light blue gradient.

**Ultra short start time, seeking time, ...**



An illustration of a person sitting on a blue sofa in a living room, watching a television. There is a coffee table in front of the sofa and a lamp on a side table.

**Advanced ABR**



A diagram illustrating Adaptive Bit Rate (ABR) streaming. It shows three horizontal film strips representing different quality levels: Q3 (top, yellow), Q2 (middle, blue), and Q1 (bottom, grey). A white line graph curves across the strips, showing the bit rate dynamically adjusting between the three quality levels.

**Video Quality**



**#1 QoE**

An icon featuring a blue star in the center, surrounded by a laurel wreath, symbolizing top performance or quality.

**Scalable over standard HTTP infrastructures**



# THANKS FOR LISTENING.

REQUEST YOUR PERSONAL ONE-ON-ONE SESSION  
WITH ONE OF OUR LOW LATENCY EXPERTS



[www.theoplayer.com/contact/low-latency-consultation-2021](http://www.theoplayer.com/contact/low-latency-consultation-2021)

**OTHER QUESTIONS? CONTACT US.**

Pieter-Jan Speelmans: [pieter-jan.speelmans@theoplayer.com](mailto:pieter-jan.speelmans@theoplayer.com)

Johan Vounckx: [johan.vounckx@theoplayer.com](mailto:johan.vounckx@theoplayer.com)