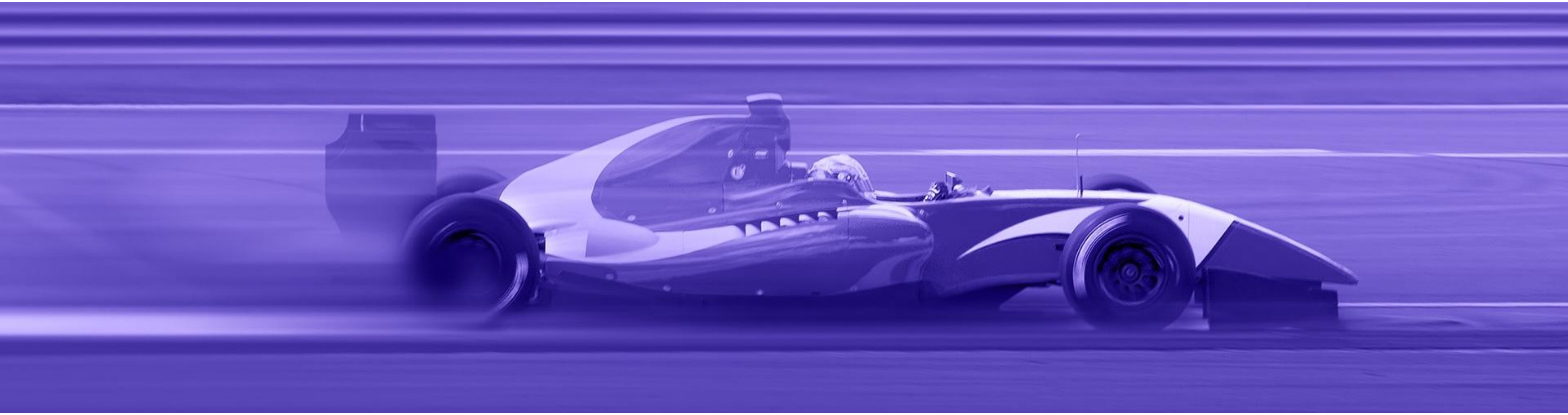
millicast

INTERACTIVE REAL-TIME STREAMING



WHIP’ING WEBRTC INTO SHAPE

WebRTC

RYAN JESPERSEN

STREAMING & DIGITAL VIDEO GEEK

RYAN@MILLICAST.COM

@RYAN_JESPERSEN

@DZNPRO

WHY AREN'T WE USING WEBRTC?

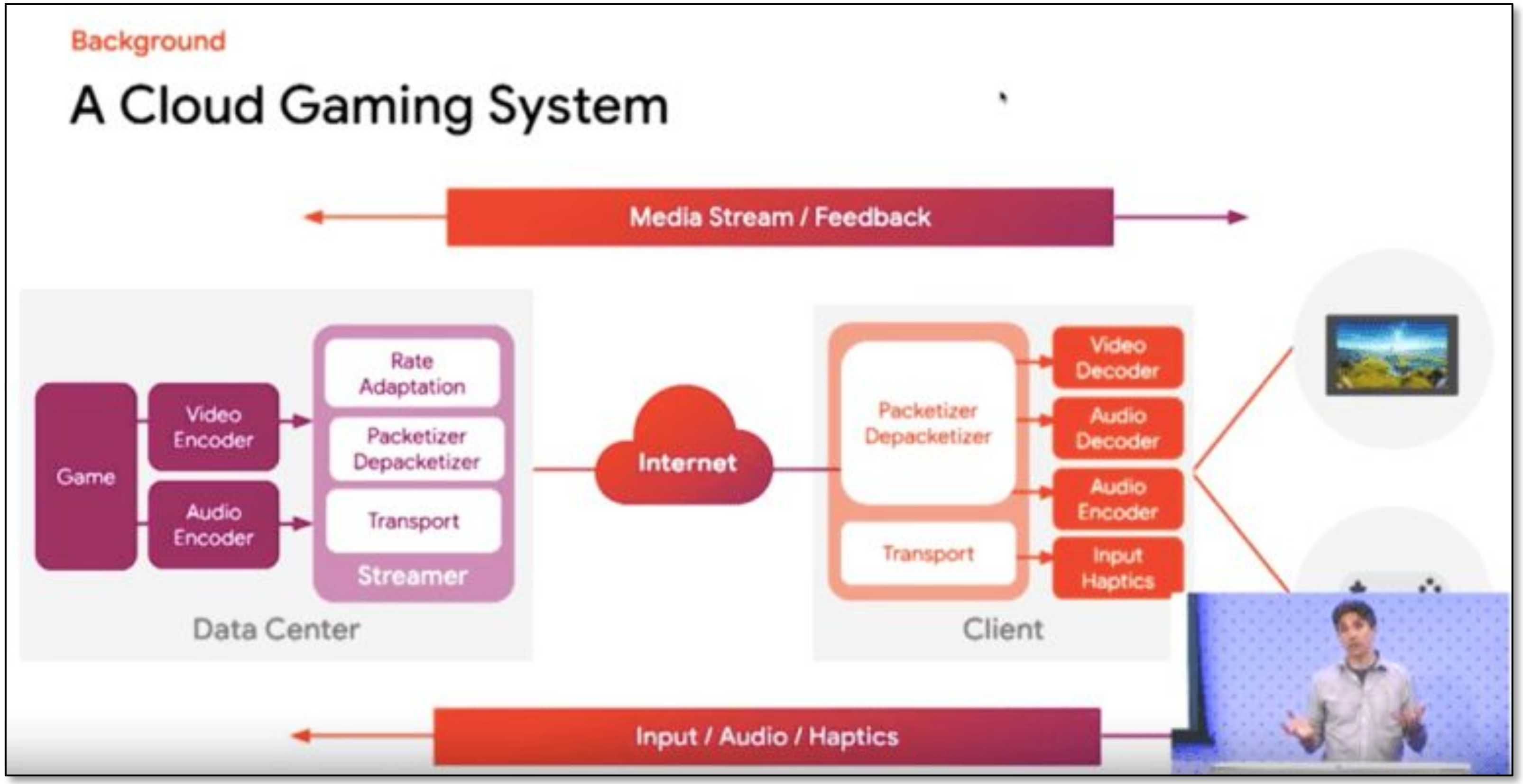
NEGATIVE PERCEPTION

WebRTC

- Because it was focused on voIP and Peer-to-Peer use cases at launch
- It's limited to a few concurrent viewers and doesn't scale
- It's associated with poor "web" quality, not for broadcast
- It requires "coding" to use

GAMING: GOOGLE STADIA

STREAMING AND MEDIA INFRASTRUCTURE



STADIA

FULL MONITORING

INTERACTIVE

E2E MEDIA LAYER

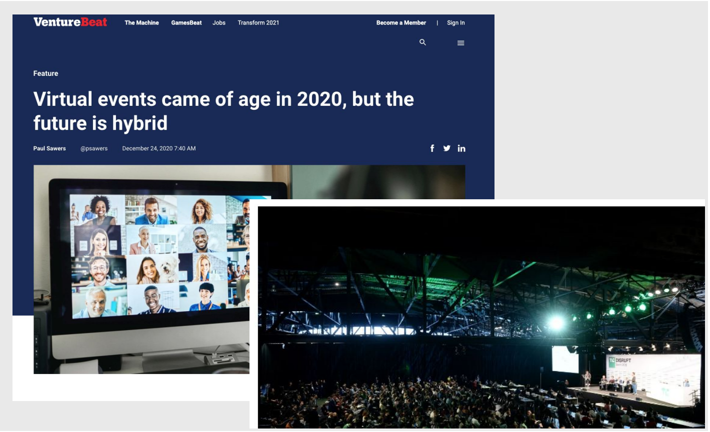
SEPARATED FROM MEDIA SERVERS

SEPARATED FROM SCALING/ROUTING

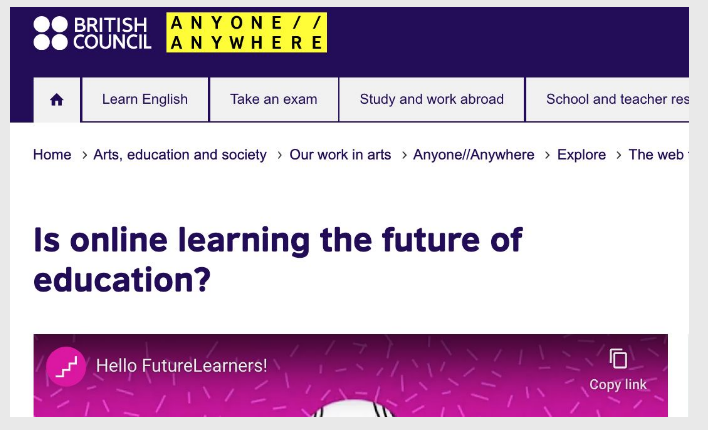
“LIVE”

5+ SECONDS

LIVE SPORTS



LIVE EVENTS



EDUCATION

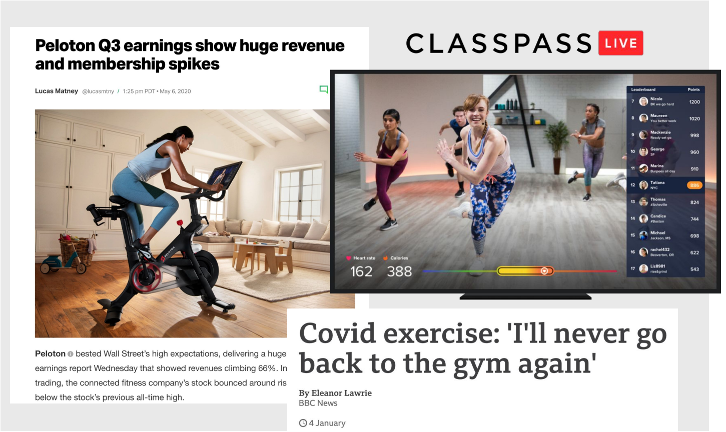
CLOUD GAMING & AR/VR

INTERACTIVITY

WILL UNLOCK VALUE IN

MANY MULTI-BILLION

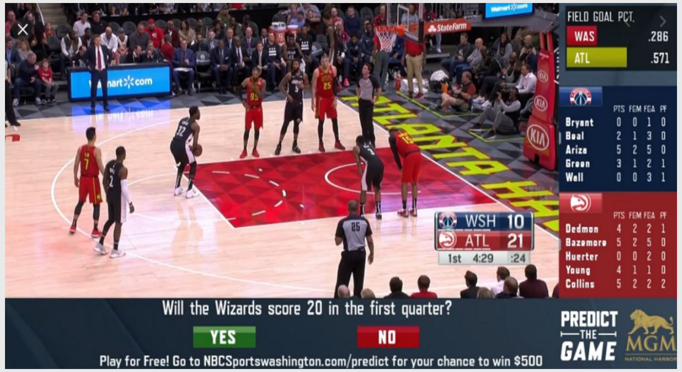
DOLLAR MARKETS



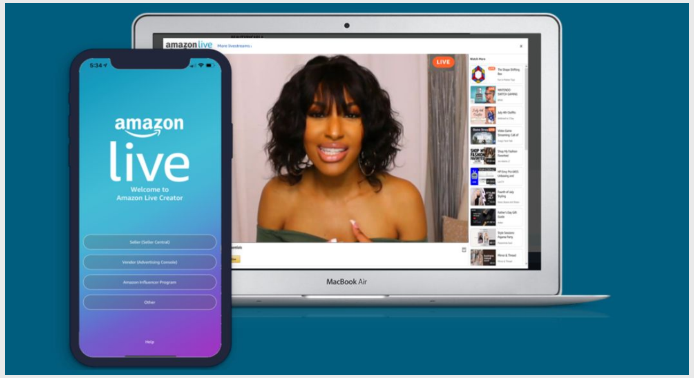
HEALTH & FITNESS

"ULTRA LOW LATENCY"

UNDER 3 SECONDS



BETTING & GAMBLING



VIDEO ENABLED E-COMMERCE

LIVE AUCTIONS

WATCH PARTIES

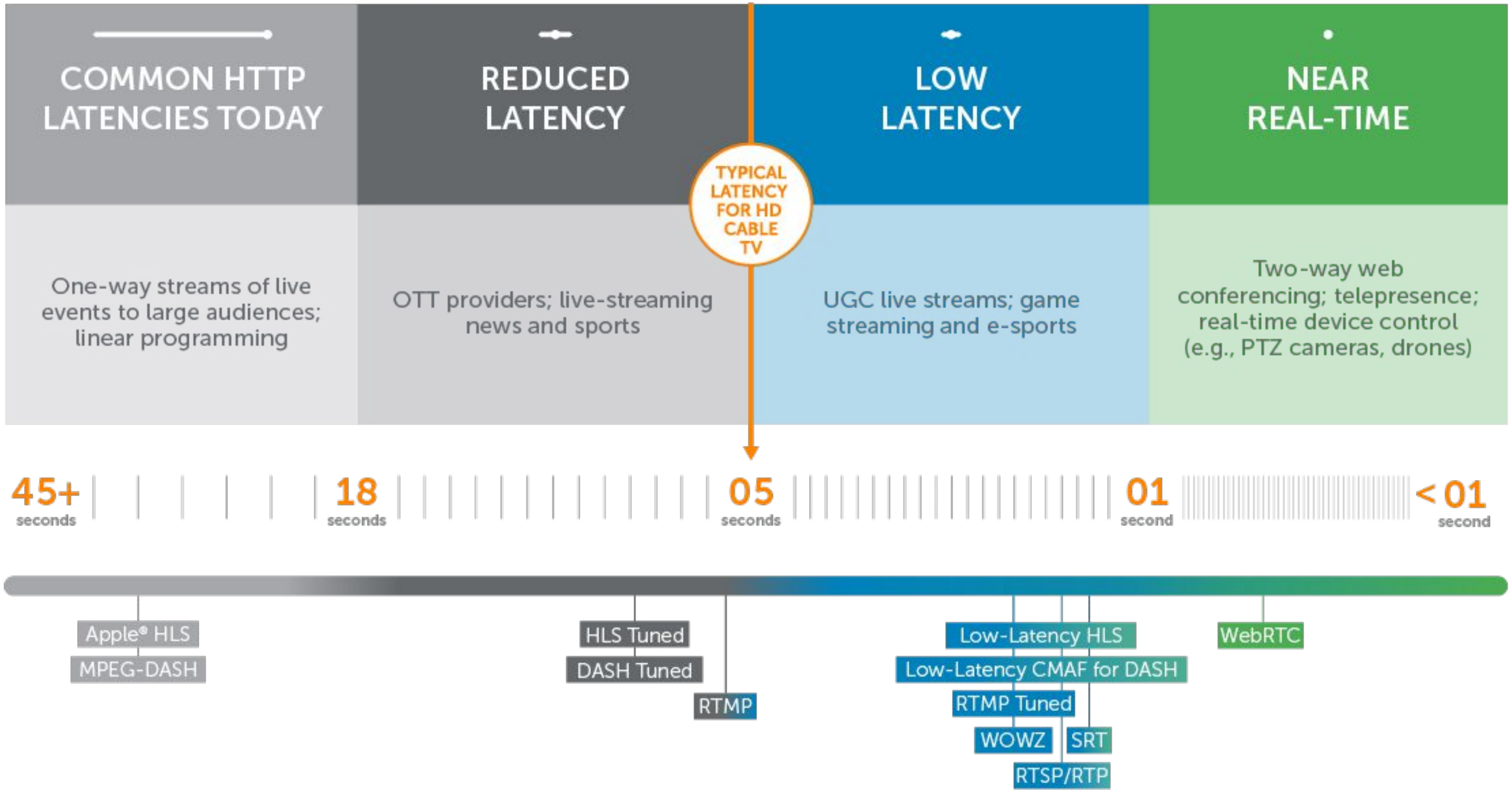
INTERACTIVITY

WILL BECOME AS
UBIQUITOUS AS EMAIL &
WEB BROWSERS

REMOTE PRODUCTION

WE CARE ABOUT LATENCY

AGAIN



WOWZA media systems

COVID ACCELERATED A MEDIA GOLD RUSH

TO INTERNET ENABLED WORKFLOWS

2020 Emmy Engineering awards were awarded to two companies providing Real-Time production streaming services.

HUMAN NATURE CRAVES *INTERACTION*

INTERACTION LEADS TO *ENGAGEMENT*

WATCH PARTIES

VIRTUAL AUDIENCE

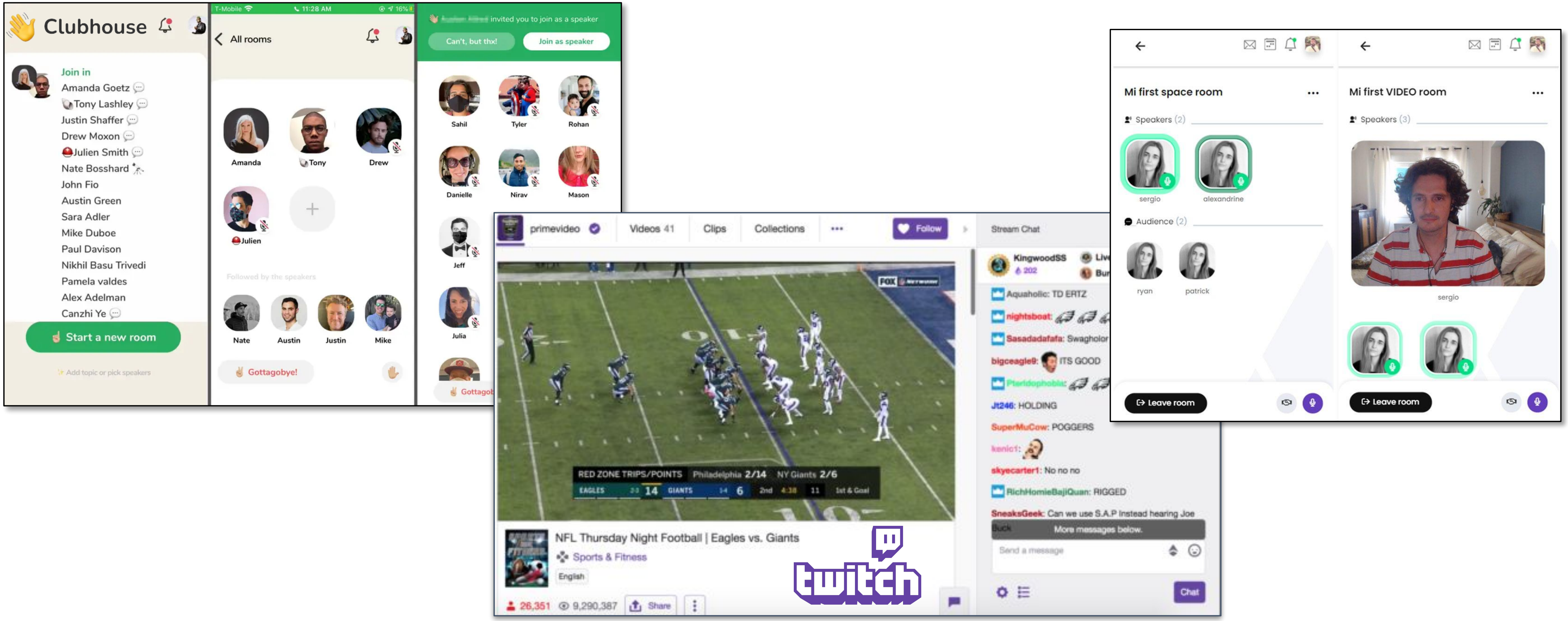
“I think it will be a big play. Digital has advantages. It has an interactive element that we can’t replicate in a linear feed. And these platforms are reaching different fans, younger fans. We want to engage all our fans.”

Roger Goodell, NFL Commissioner
CNBC, Squawk Alley
September 2nd, 2020

REMOTE **NEEDS** TO **FEEL REAL**

HAVE WE FINALLY BRIDGED THE GAP

BETWEEN WEB & BROADCAST?



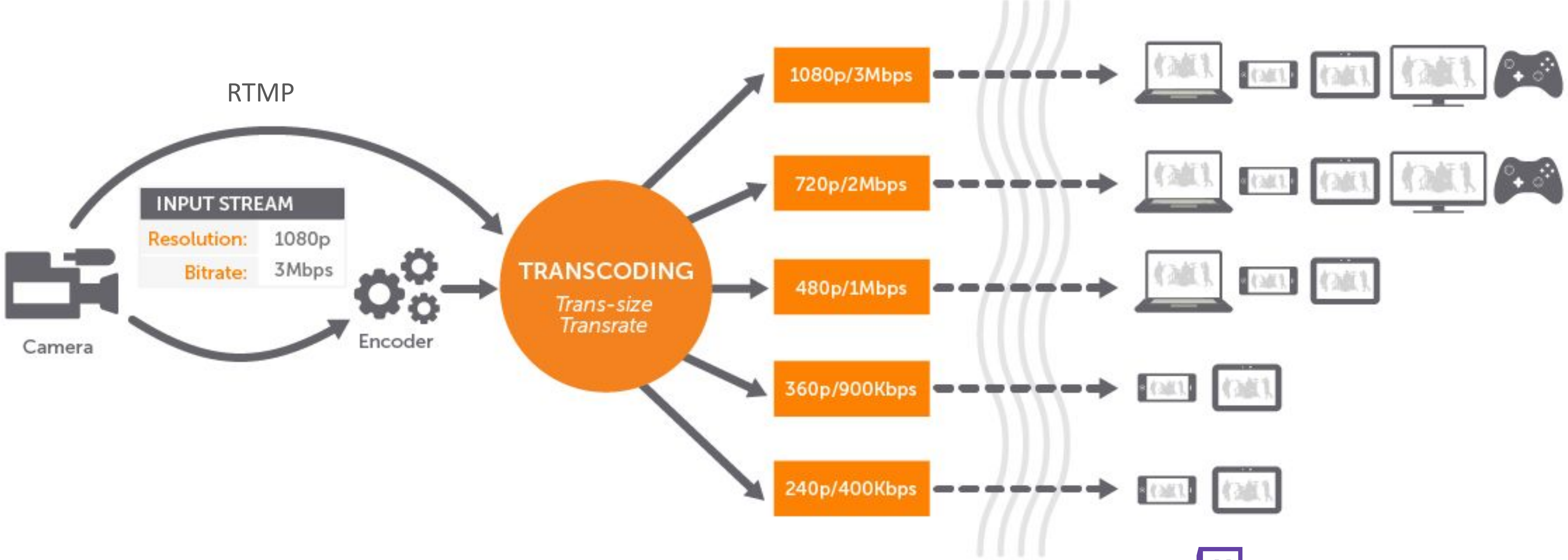
WEBRTC IS NOT "COMPLETE"

WE NEED TO TALK

- There is no standard signaling protocol available to pair with WebRTC:
    - SIP or XMPP are not designed to be used in broadcasting/streaming.
    - RTSP, which is based on RTP and maybe the closest in terms of features, is not compatible with WebRTC SDP offer/answer model.
- Each WebRTC streaming service requires implementing a custom ad-hoc protocol.

RTMP IS STILL UBIQUITOUS

FOR LIVE STREAMING



MUX · YouTube Live · vimeo · livestream · twitch · WOWZA media systems

WE NEED BROADCAST-QUALITY FEATURES

WITH CONSUMER-GRADE WORKFLOWS

WebRTC

SVC

NETWORK RESILIENCE

E2EE

CONTENT PROTECTION

HDR 10

BROADCAST QUALITY

AV1

REAL-TIME ENCODING

LET'S EMBRACE INTERNET, WEB & MEDIA STANDARDS

TO REACH EVERY DEVICE

WebRTC

IETF

W3C MEMBER

Google

Firefox

Apple

Microsoft

IETF.ORG/BLOG/WEBRTC-STANDARDIZED/

W3.ORG/TR/WEBRTC/

I WANT “REAL-TIME"

BUT CAN I BRING MY TOYS?

Open Broadcaster Software

Wirecast

vMix

Decklink SDI 4K

Decklink 8K Pro

Decklink 4K Extreme 12G

Decklink Mini Monitor 4K

WE NEED A REFERENCE SIGNALLING PROTOCOL

RIGHT NOW!

- Must be simple to implement and as easy to use as an RTMP URI.
- Support ingest, which is a subset of possible WebRTC use cases:
  - Only needs to support unidirectional flows.
  - No need to support renegotiations.
  - Server is assumed to be a public IP or deployed in same private network as publisher
- Fully compliant with WebRTC and RTCWEB specs for the given use case.
- Must support authentication.
- Usable both in web browsers and in native encoders.
- Supports load balancing and redirections.
- Lower the requirements for both hardware encoders and broadcasting by reducing non-essential features.

OBS STUDIO WEBRTC

WEBRTC END-TO-END

CoSMoSoftware/**OBS-studio-webrtc**

This is a fork of OBS-studio with generic support for webrtc. It leverages the same webrtc implementation most browsers use.

100 Contributors · 21 Issues · 398 Stars · 103 Forks

GITHUB.COM/COSMOSOFTWARE/OBS-STUDIO-WEBRTC

OBS STUDIO WEBRTC

WEBRTC END-TO-END

OBS
WEBRTC

WebRTC → Medooze → WebRTC

iOS

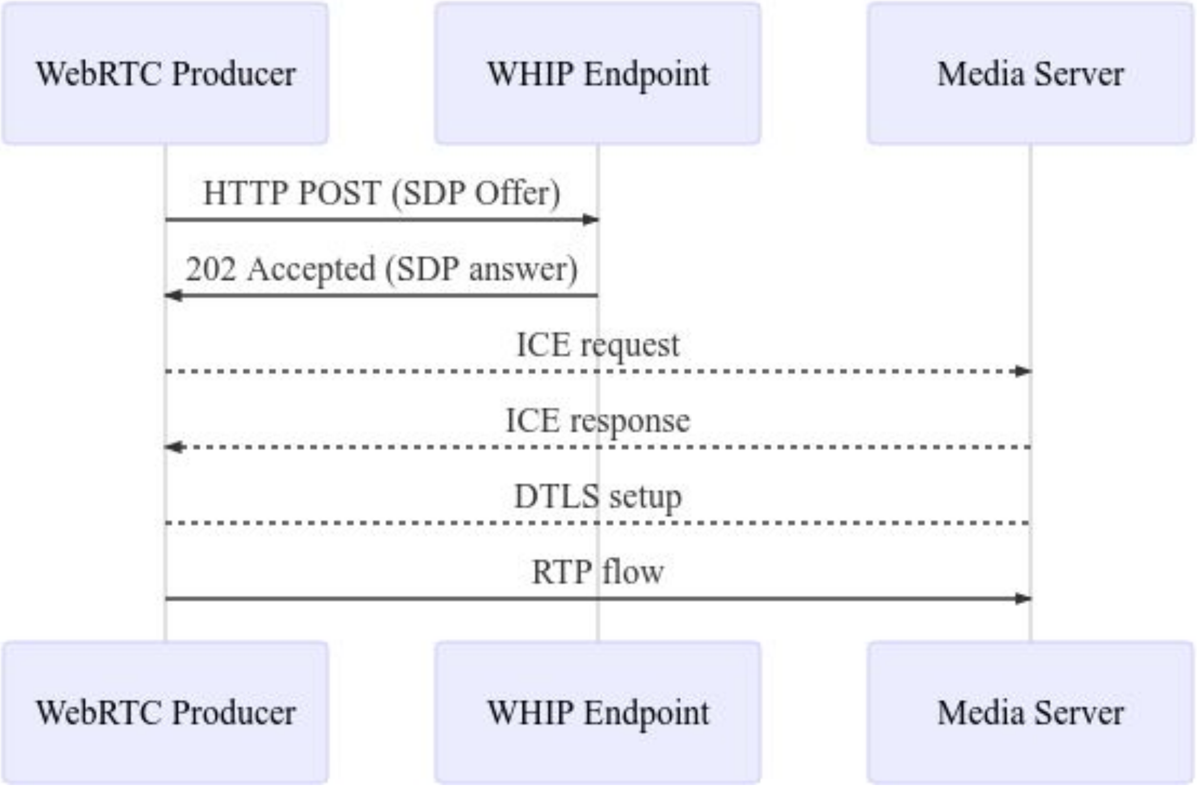
GITHUB.COM/COSMOSOFTWARE/OBS-STUDIO-WEBRTC

WHIP (WEBRTC HTTP INGESTION PROTOCOL)

THE MAGIC BULLET FOR ENCODERS

PROPOSED SOLUTION

WHIP



TOOLS.IETF.ORG/HTML/DRAFT-MURILLO-WHIP-00

PROPOSED SOLUTION

WHIP

- HTTP POST for exchanging and SDP O/A.
- Connection state is controlled by ICE/DTLS states:
  - ICE consent freshness [RFC7675] will be used to detect abrupt disconnection.
  - DTLS teardown for session termination by either side.
- Authentication and authorization is supported by the Authorization HTTP header with a bearer token as per [RFC6750].
- Support HTTP redirections for load balancing.

WHIP (WEBRTC HTTP INGESTION PROTOCOL)

THE MAGIC BULLET FOR ENCODERS

WHIP is a way to standardize the WebRTC signaling layer and establish the WebRTC connection using a simple HTTP request/response. It's already in:

MILLICAST STUDIO

OBS WEBRTC

Medooze

JANVS WEBRTC SERVER

But broadcasters still want hardware for physical SDI and HDMI capture.

TOOLS.IETF.ORG/HTML/DRAFT-MURILLO-WHIP-00

SIMPLE JAVASCRIPT CLIENT

WHIP

medooze/**whip-js**

Simple WHIP client javascript module

2 Contributors	0 Issues	10 Stars	2 Forks

GITHUB.COM/MEDOOZE/WHIP-JS

SIMPLE JAVASCRIPT CLIENT

```
//Get user media
const stream = await
navigator. mediaDevices .getUserMedia ({audio :true, video
:true});

//Create peer connection
const pc = new RTCPeerConnection ();

//Listen for state change events
pc.onconnectionstatechange = (event) =>{

        switch (pc.connectionState )
        {
                case "connected"
                :
                        break
                        ;
                case "disconnected" :
                        break ;
                case "failed"
                :
                        break
                        ;
                case "closed"
                :
                        break
                        ;
        }
}
```

```
//Send all tracks
for (const track of stream. getTracks ())
        //You could add simulcast too here
        pc.addTrack
        (track);
//Create SDP offer
const offer = await pc.createOffer ();
await pc.setLocalDescription (offer)
//Do the post request to the WHIP endpoint with the SDP offer
const fetched = await fetch (url, {
        method : "POST"
        ,
        body: offer.
        sdp,
        headers
        :{
                "Content-Type" :
                "application/sdp"
        }
});
//Get the SDP answer
const answer = await fetched. text();
await pc.setRemoteDescription ({type:"answer" ,sdp:
answer});
```

gstreamer

WHIP

lminiero/**simple-whip-client**

Simple WHIP Client (based on GStreamer's webrtcbin)

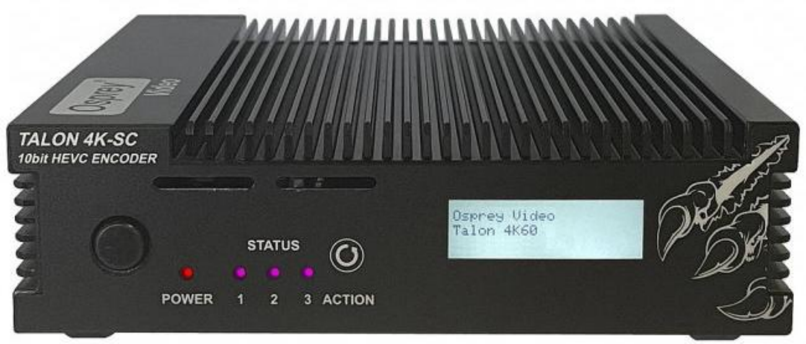
2 Contributors · 0 Issues · 3 Stars · 0 Forks

GITHUB.COM/LMINIERO/SIMPLE-WHIP-CLIENT

OSPREY VIDEO

TALON 4K-SC

Osprey® Video



WebRTC

H.265 | H.264

12G-SDI and HDMI 2.0
4096x 2160p 10bit 4:2:2
16/8 Audio Channels

OSPREYVIDEO.COM/TALON-ENCODERS

WHAT'S NEXT?

WHIP'ING WEBRTC INTO SHAPE

WEBRTC INTEGRATION GUIDE

THE FUTURE FOR ENCODERS

Below are non-mutually exclusive options to add WebRTC to these encoders:

1. Ground Zero

Implement a super low latency version of your RTMP encoder. It's really just playing with RTMP parameters, nothing fancy, minimum overhead. You will not get ABR, you will not get E2EE, you will not get a better codec than H.264, but it will work with the Millicast platform today as an RTMP ingest. Frankly speaking, that's investing in the past.

2. Implement WebRTC+WHIP with H264 or VP8 (4:2:0, 8bits)

This is the most sensible first step. It's simple to implement, it will work against all existing browsers today, it will shave half of the latency you have with RTMP. There is already a night and day difference for latency-sensitive workflows using RTMP today. It allows you to implement and validate a full WebRTC stack, before you move on to further integration. You will not get ABR, you will not get E2EE.

3. Intermediary: WebRTC+WHIP with VP9 mode 2 (10bits 4:2:0 HDR)

An interesting intermediate step if your hardware supports VP9 encoding (INTEL, Qualcomm and Samsung do for example). This provides you with a 10bits HDR10 capacity out of the box, supported by Chrome, Edge and Safari today.

4. Intermediary: WebRTC+WHIP with H.265

This is kind of a smaller play. Only Apple Safari will be able to receive your stream and display it among existing browsers (and it's not likely to change). However, there are also a lot of Hardware devices that can decode H.265. Risky and not very practical, but many existing devices support H.265. It is low hanging fruit, and god knows Apple owners love their devices.

5. Same as #2 but with simulcast

This gives you the best quality possible today, while being future-proof when E2EE will be available. In our opinion, this is the best configuration of the base offer (in-par with browsers today). It requires the capacity for multiple concurrent encodings. Note that only one stream is high resolution, and all other streams will be lower resolutions. In this context, the Qualcomm approach with some CPUs/GPUs more capable than others make a lot of sense. The magic number is 3 encoders in parallel for optimum quality.

6. Real-Time AV1 SVC, or other high level codecs (i.e. Dolby Atmos, etc.)

There will always be a demand for the best quality possible: 12bits, 4:4:4, lossless (no quantization, etc). This will be our premium offer. AV1 is very interesting because of its widespread adoption on the decoder side, and the fact that encoders will find their way into the browsers very soon. Also, there are many very good libraries implementing the codec already making adoption easier.

MILLICAST.MEDIUM.COM/WEBRTC-THE-FUTURE-FOR-ENCODERS-8C08448C16E9

WHAT’S NEXT?

WHIP’ING WEBRTC INTO SHAPE

Broadcast Quality:

- 4:4:4 colour
- 10-bit & 12-bit
- HDR & Dolby Vision
- 5.1 & 7.1 Surround Sound

Device Reach:

- Apple iOS
- Apple tvOS
- Chromecast

Security:

- End to end encryption (Sframe)
- Forensic Watermarking

Resiliency:

- AV1 SVC (Scalable Video Coding)
- Sender & Receiver Analytics

REMOTE PRODUCTION

USE CASE

BirdDog | CLOUD

NBC
prime video
avp

"After we went through all the emotions of losing our full season, we started going through how we were going to bring the sport to the world during the COVID crisis."

Josh Glazebrook, AVP Creative Director

millicast
The Fastest Streaming on Earth

BirdDog

THE PROBLEM

When delivering elite live beach volleyball to global fans around the world, Amazon Prime and NBC turned to the Millicast real time streaming service and BirdDog IP cameras to ensure their producers have a high-quality, low latency live feed for remote production.

AVP is the premiere Pro Beach Volleyball league featuring the world's elite players.

In 2020, AVP, like many sports organizations, had to postpone it's competition schedule due to stay-at-home orders.

HELLO.MILLICAST.COM/LIVE-REMOTE-PRODUCTION-CASE-STUDY-PRIME-VIDEO-NBC-AND-AVP

NFL

PACIFIC POST

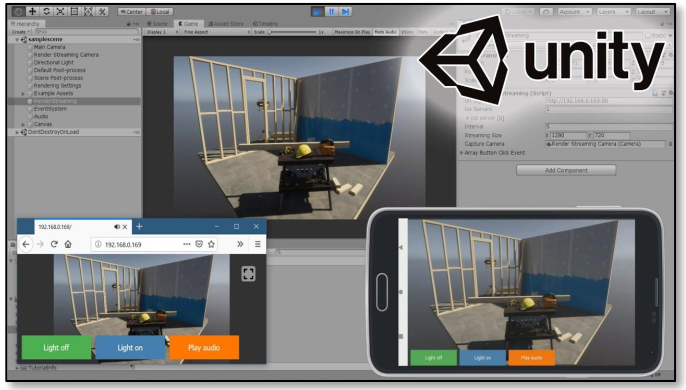
SetStream.io

UNITY GAME ENGINE

END-TO-END WEBRTC WORKFLOW

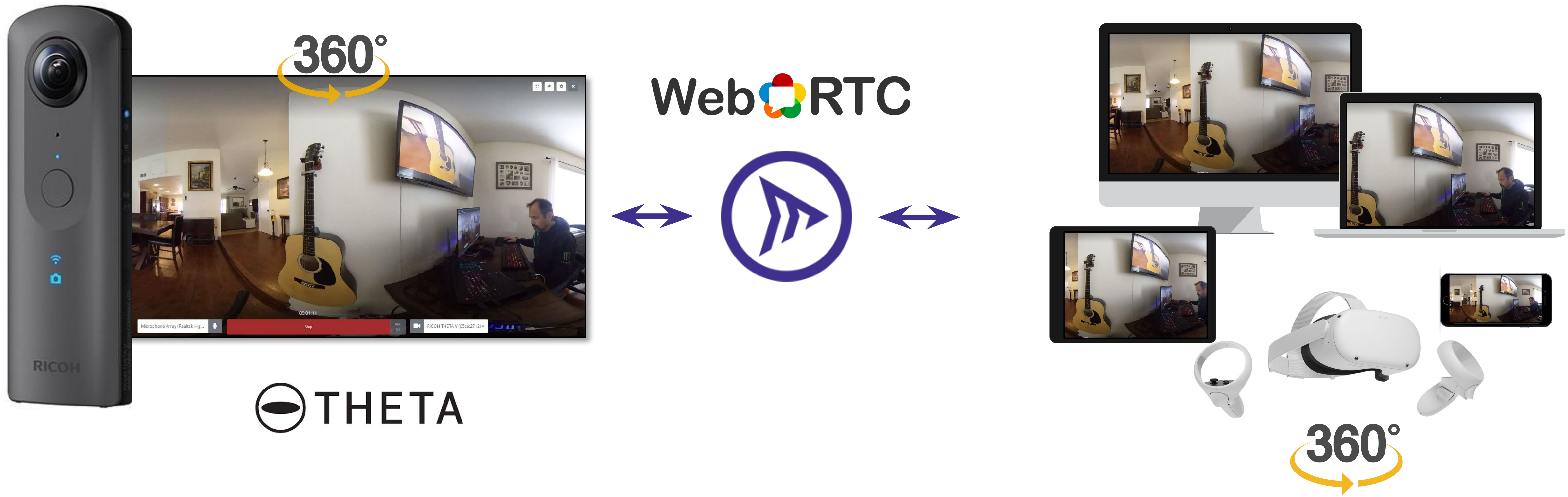
WebRTC unity

WebRTC



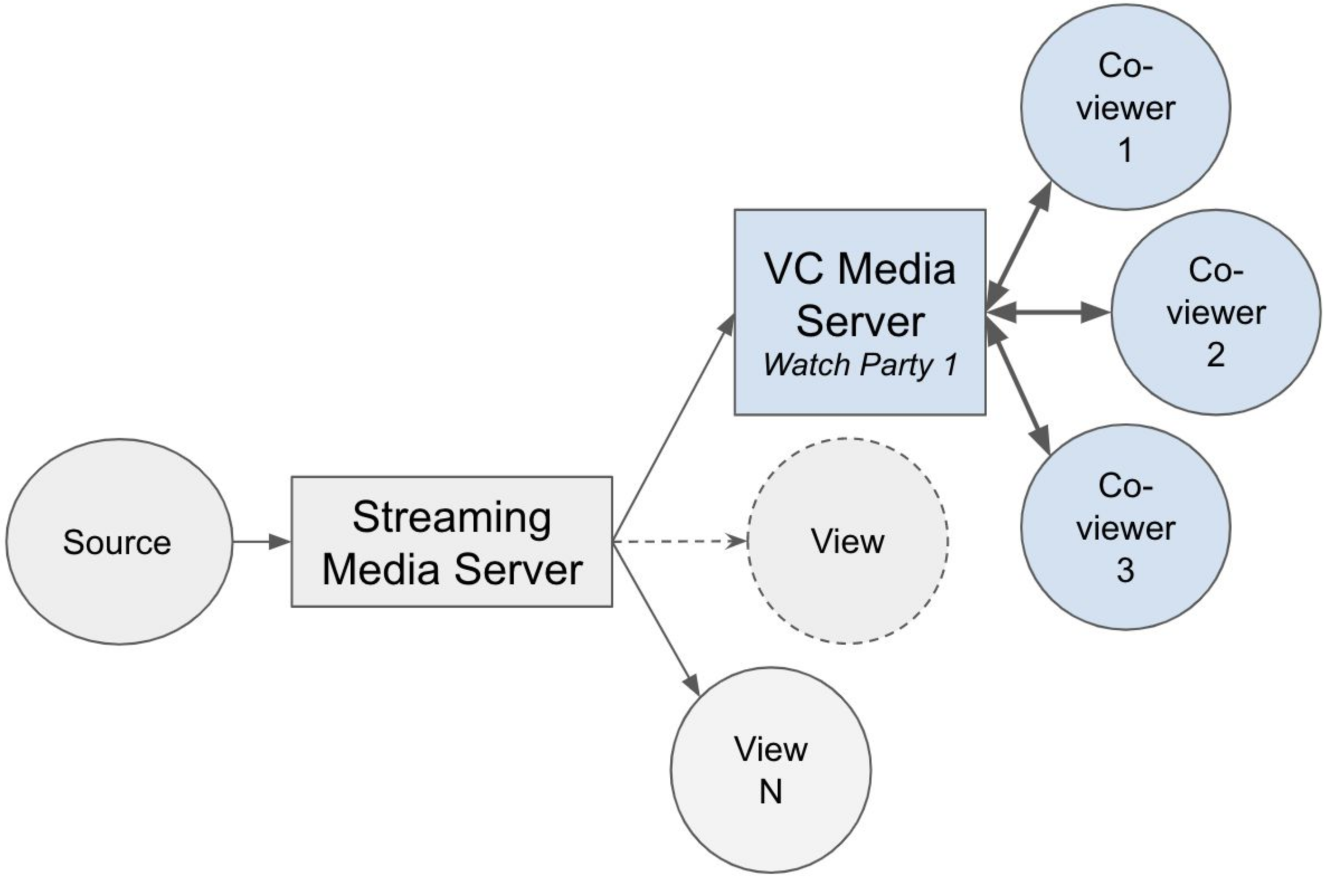
VR & 360° LIVE STREAMING

WEBRTC WORKFLOW



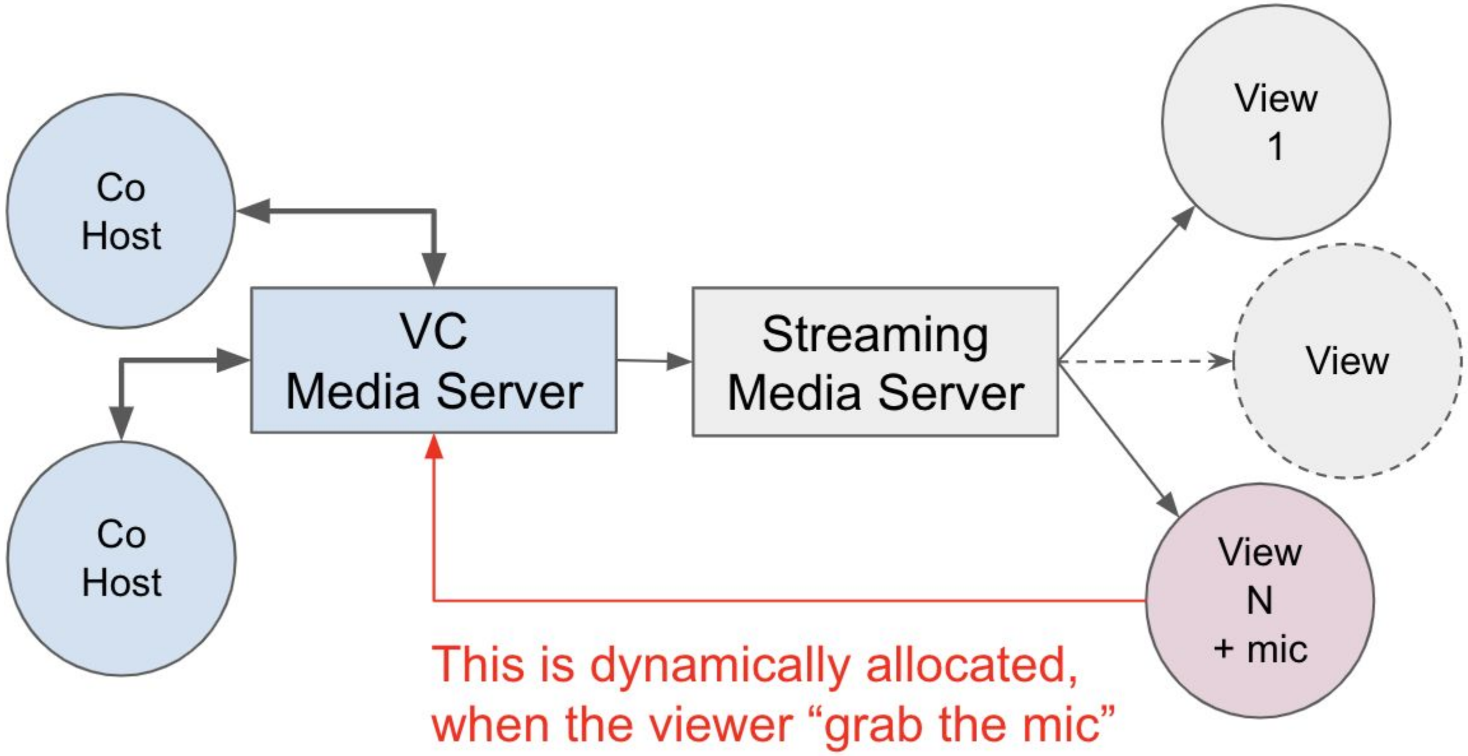
INTERACTIVE MEDIA

WATCH PARTIES



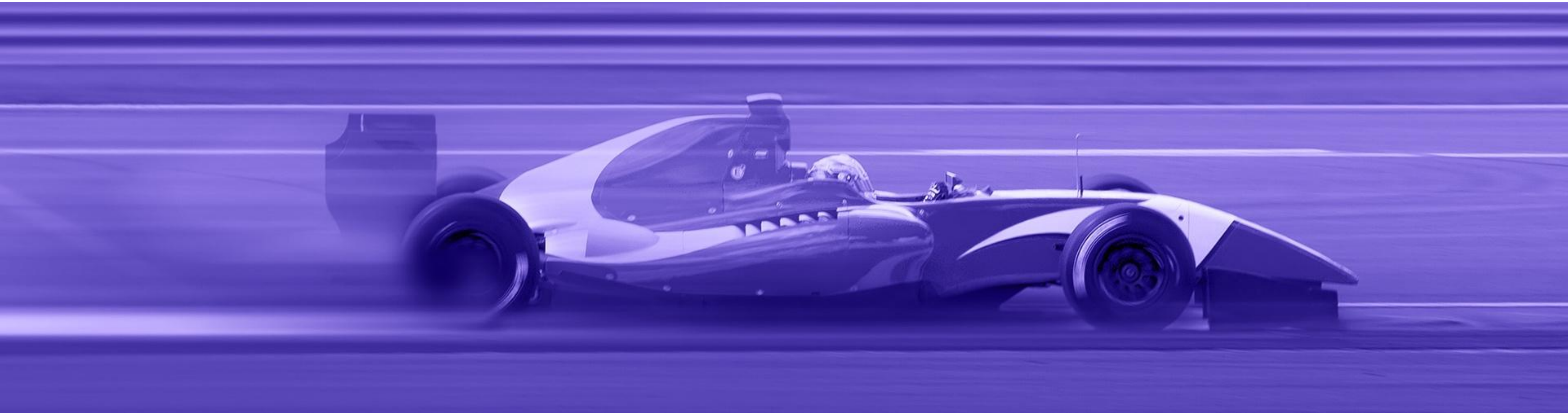
INTERACTIVE MEDIA

STREAM PARTIES



millicast

INTERACTIVE REAL-TIME STREAMING



WHIP'ING WEBRTC INTO SHAPE

WebRTC