

Merging Two Worlds - Broadcast and WebRTC

WHIP and WHEP

Dan Jenkins / FOSDEM 2023



Dan Jenkins

**WebRTC & Real Time Media
Geek**

dan@everycastlabs.uk

@dan_jenkins

@danjenkins@fosstodon.org

Founder -

Everycast Labs Ltd



Merging Two Worlds

Broadcast and WebRTC

**Firstly,
a few
definitions...**

WebRTC

Web Real-Time Communication

WebRTC

- Encrypted by default
- Sub-second glass to glass
- Open Source
- Two-way communications
- No signalling defined in the spec
- Required Open Source Codecs
- Embedded in every browser
- Use your own codecs in your own implementations
- Delivery over UDP
- NAT busting with ICE
- Many libwebrtc independent versions available now

**"No signalling
defined in the spec"**

It was a good thing

SRRT

Secure Reliable Transport

SRT

- Open Source
- Used heavily in the broadcast industry
- UDP based
- Requires native apps
- Can be encrypted optionally
- Codec agnostic
- Can be sub-second but usually multiple seconds for reliability
- Can be used across the internet or wide area networks or inside a LAN

NDI

**Network Device Interface
(what a generic name)**

NDI

- Not Open Source
- Comes in multiple forms - NDI, HX, HX2 and HX3
- Designed to work within a LAN
- Not open source - but free to use
- Uses UDP
- Closed Source SDK
- Licensing is "more complicated"

RIST

Reliable Internet Stream Transport

RIST

- Open Source
- UDP based
- Encrypted
- RTP based
- Relatively new compared to the other options
- Designed to work over the internet (WAN) or LAN

The other forms of
media transport
aren't really what
I'd call real-time

Merging **two** worlds

WebRTC and Broadcast

Broadcast and WebRTC

**They can finally
live in harmony**

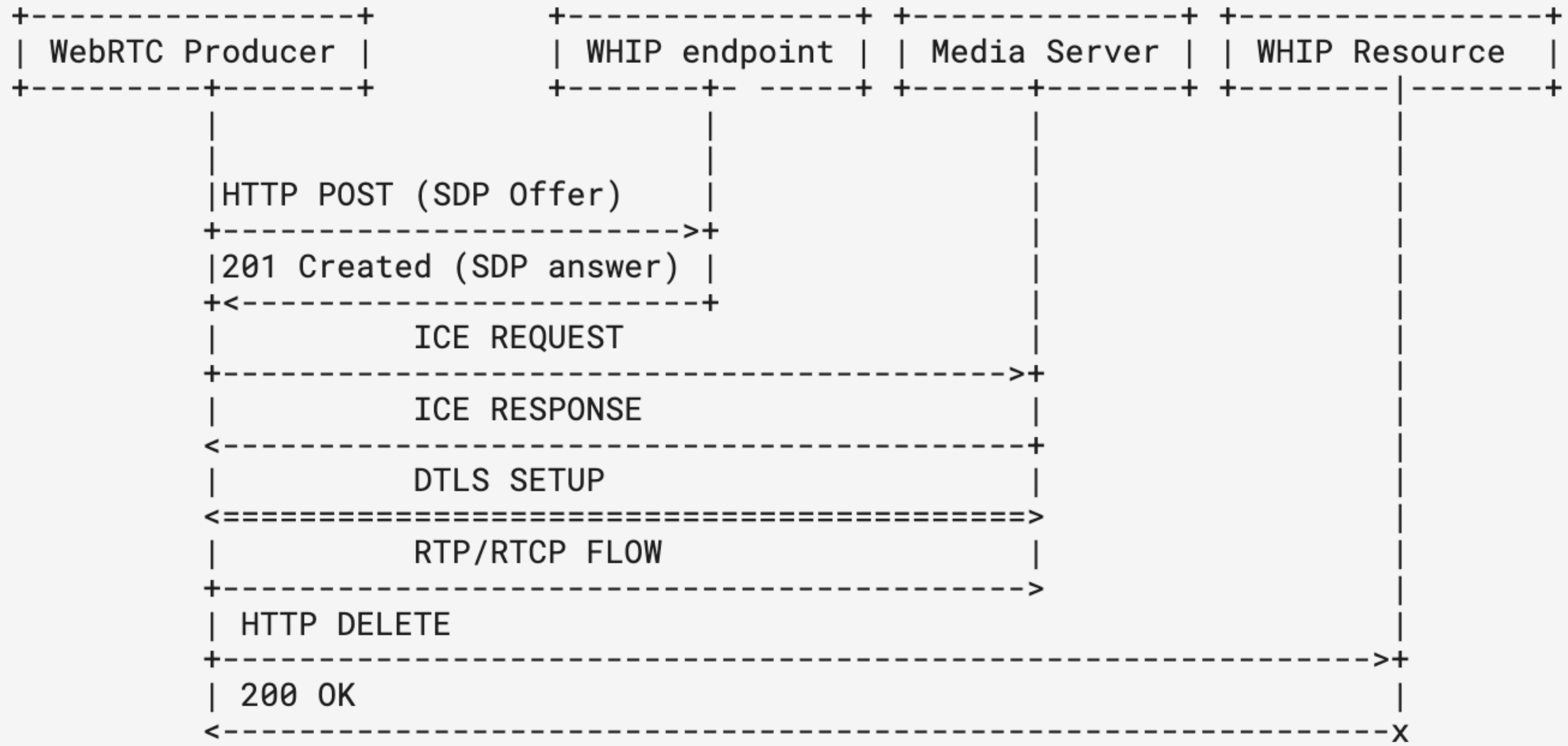
Maybe.

WHIP and

WHEP

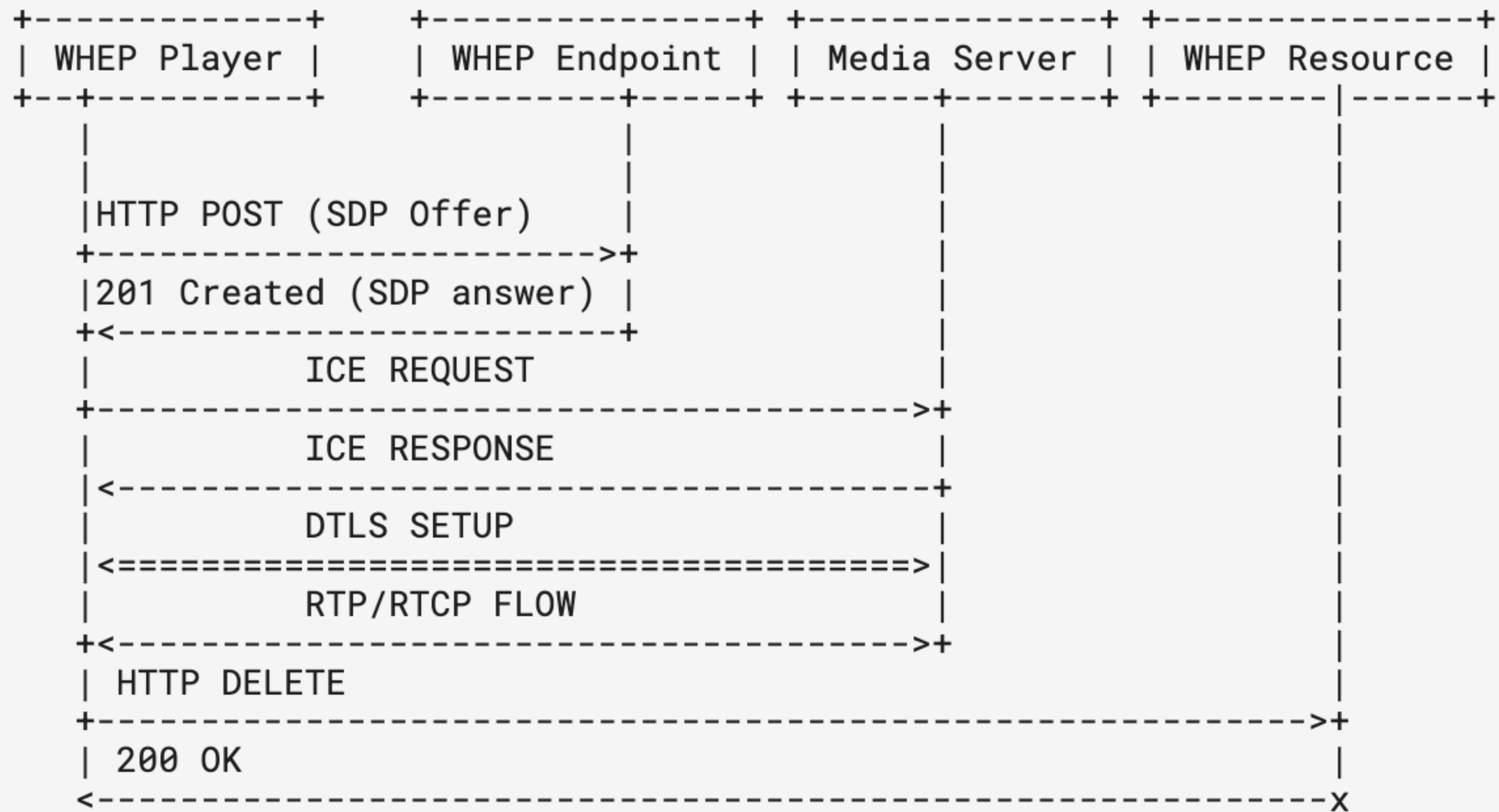
WHIP

WebRTC HTTP Ingestion Protocol



WHHEP

WebRTC HTTP Egress Protocol



WHOA

WebRTC HTTP Offer Answer Protocol

WHOA

~~WebRTC HTTP Offer Answer Protocol~~



W H O A

Those look like signalling protocols to me...

**WebRTCs lack of
enforced
signalling was
great...**

Until it wasn't

No enforced
signalling protocol
meant lack of
industry support

**How do you use
WebRTC to deliver
media while
implementing a different
API for each provider?**

You don't

You use
<insert other
protocol here>

**Whether you're
a fan of WebRTC
or not...**

It has its uses.

And up until
recently... **difficult to**
use outside of a
service with an **SDK**

**Interop was
plain difficult.**

WHIP

and

WHEP

WHIP

WebRTC HTTP Ingestion Protocol

WHEP

WebRTC HTTP Egress Protocol

WHIP

**WebRTC HTTP ~~Ingestion~~ Protocol
Ingress (??)**

WHEP

WebRTC HTTP Egress Protocol

Both drafts in the IETF

<https://datatracker.ietf.org/doc/draft-ietf-wish-whip/>

<https://datatracker.ietf.org/doc/draft-murillo-whep/>

**So what are
they?**

WHIP

Do **HTTP POST** with **SDP offer**,
Get **SDP answer** in response.
Done.

Sending media to server

WHIEP

**Do HTTP POST with SDP offer,
Get SDP answer in response.
Done.**

Receiving media from server

OK...

But what does that get us?

Hardware Encoder/Decoders

TALON HARDWARE ENCODERS

TALON 4K-SC

12G-SDI | HDMI2.0

10-bit 4:2:2

4096 x 2160P60

Status Display

16 Audio Channels SDI | 8 Audio Channels HDMI

H.265 HEVC | H.264 AVC ENCODING



TALON UHD-SC

12G-SDI

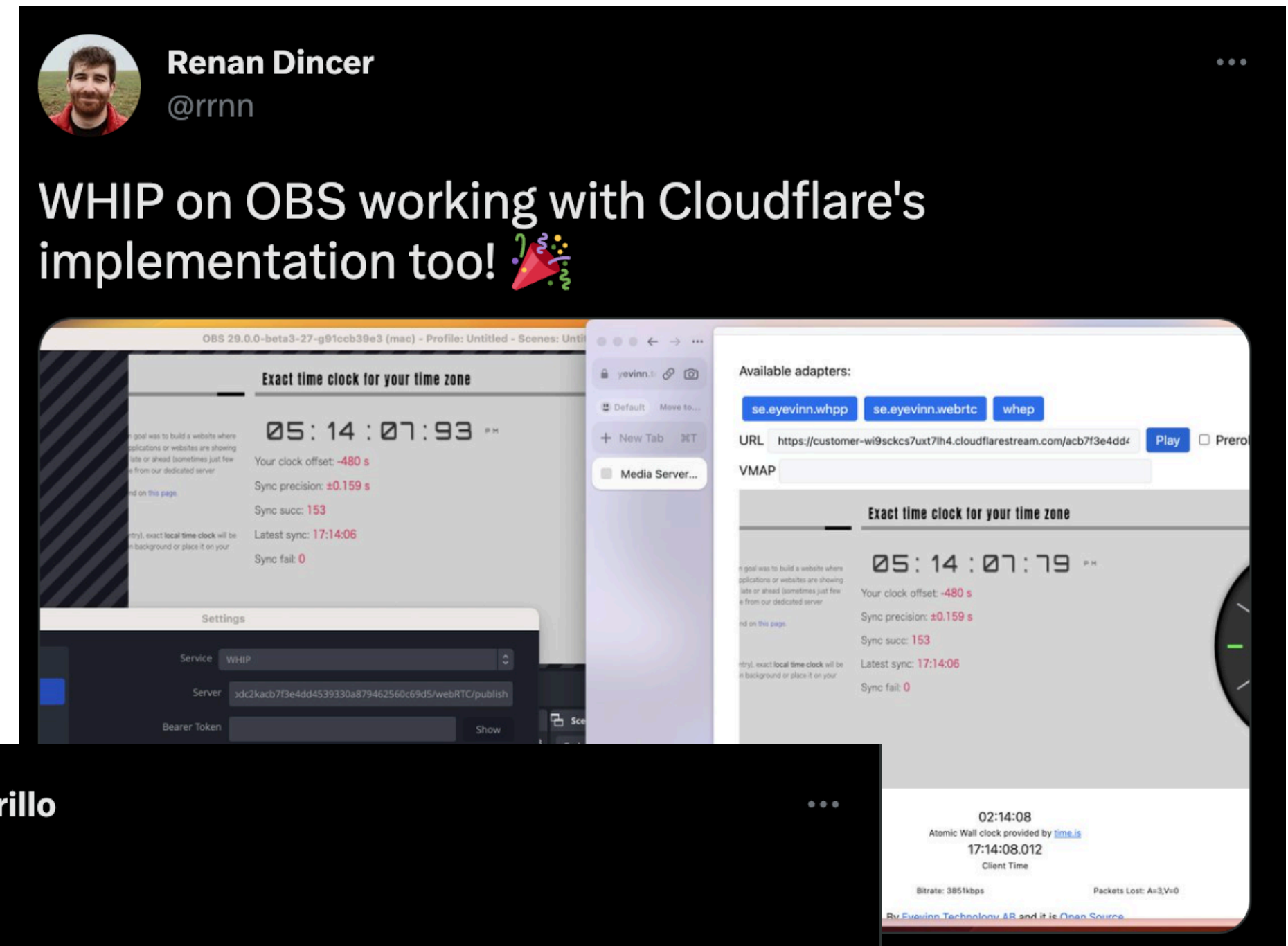
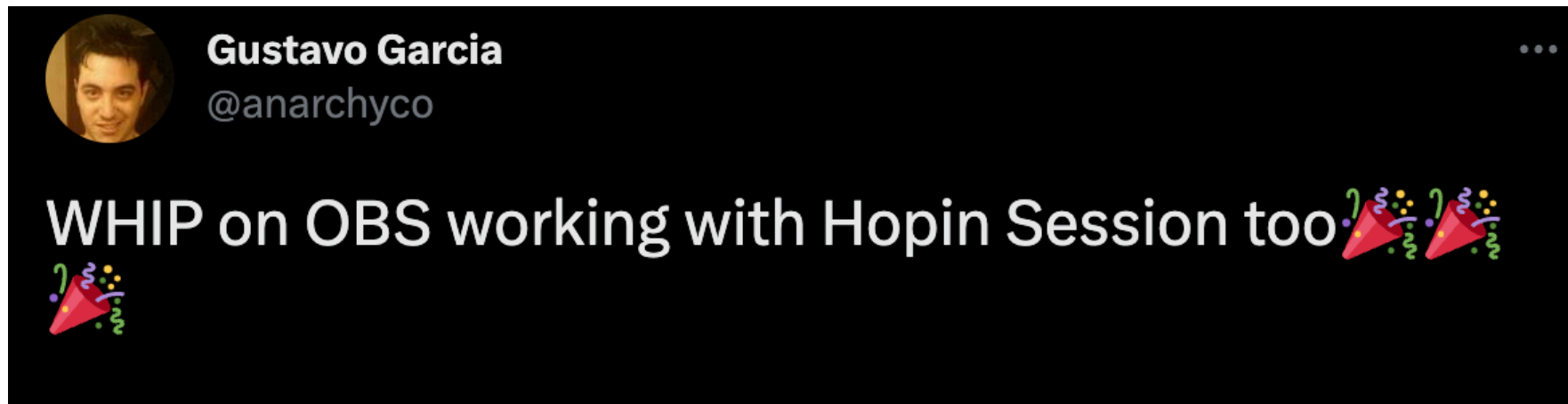
10-bit 4:2:2

3840 x 2160P60

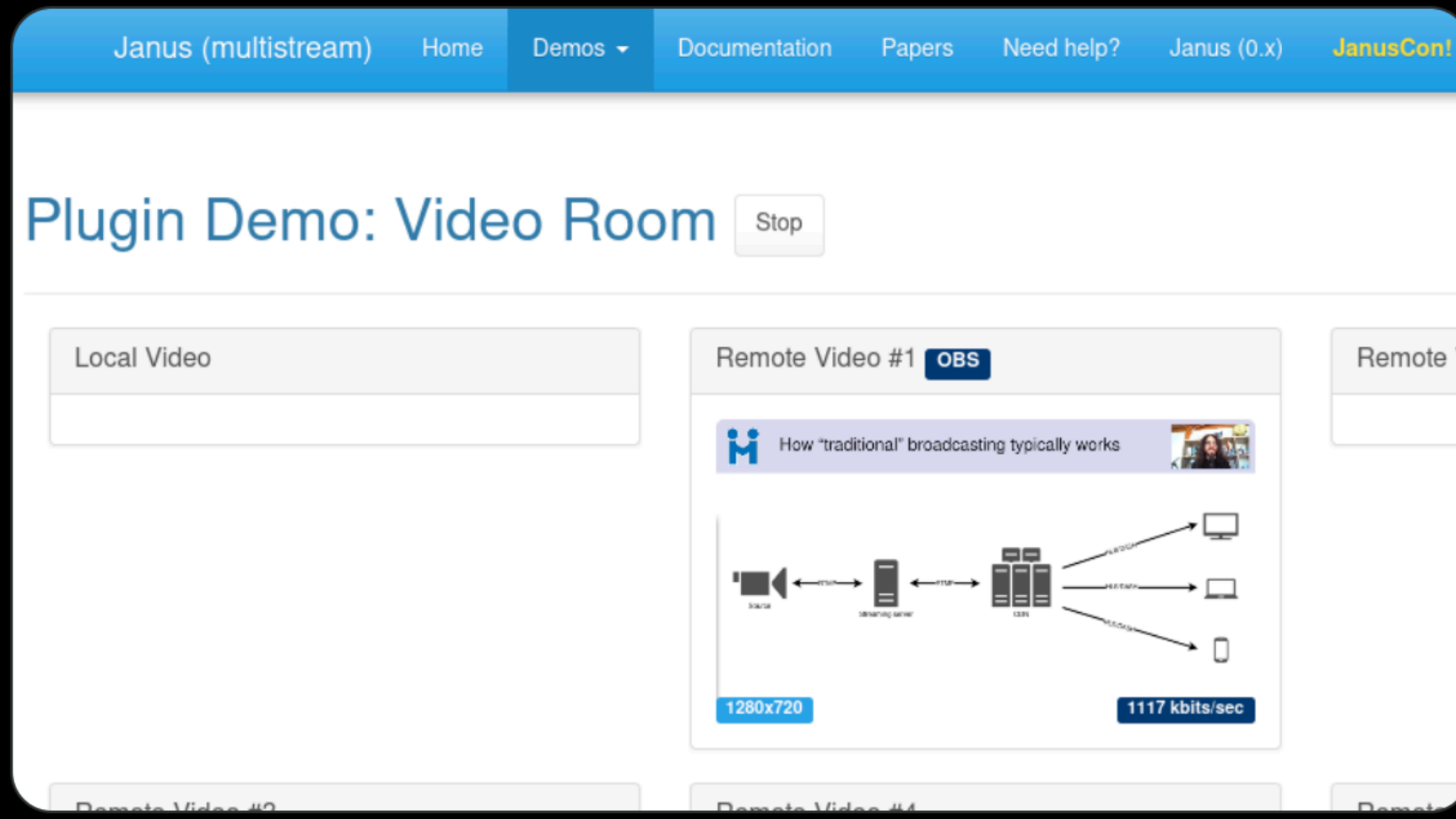
16 Audio Channels SDI

H.265 HEVC | H.264 AVC ENCODING

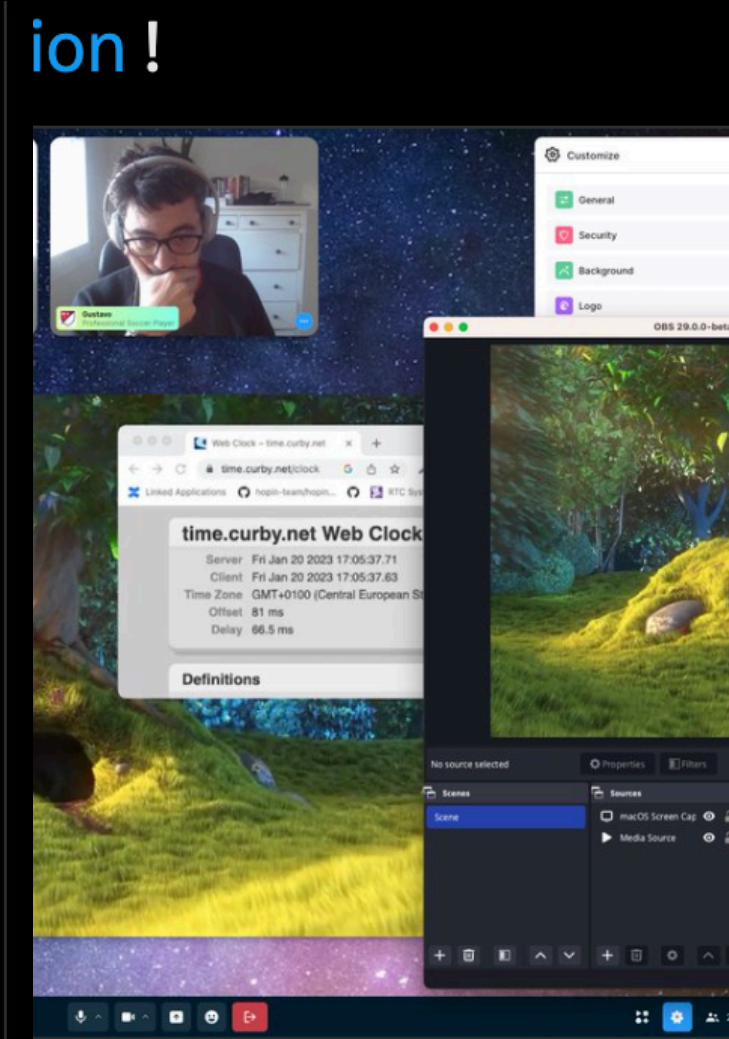
Software support - OBS



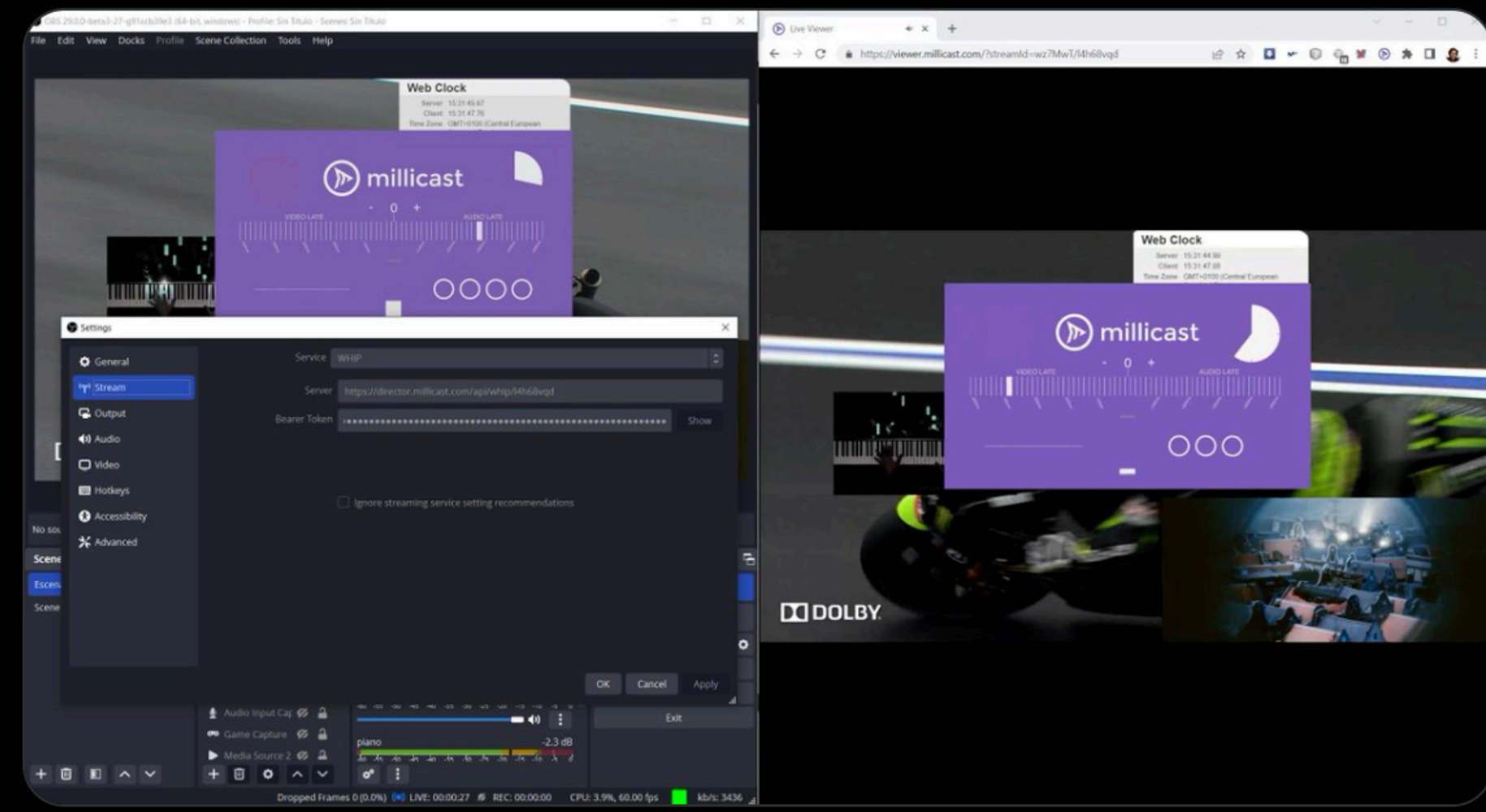
#WHIP support in #OBS is coming up nicely! This is a screenshot of OBS publishing to a #Janus VideoRoom instance using my WHIP server 🙌 The only quirk I noticed so far is negotiation, since OBS seems to be offering many codecs but only H.264 is (currently?) implemented



12:41 PM · Jan 18, 2023 · 2,758 Views




Replying to @elminiero
working with millicast too! nice!



Software support - GStreamer

WhipSink: A bin for WHIP

Code ▾

 Merged Taruntej Kanakamalla requested to merge [tkanakamalla/gst-plugins-r...](#) into [main](#) 5 months ago

All threads resolved! ⋮

Overview 44 Commits 1 Pipelines 15 Changes 9

Working version with asynchronous HTTP calls. Performs an HTTP request to exchange SDP offer/answer with a given endpoint which supports WHIP. Once the local and remote description are set to the webrtcbin (child element) the stream gets added as send-only (unidirectional)

Example Pipeline:


```
gst-launch-1.0 filesrc location=~/.Videos/sintel_trailer-1080p.mp4 !
decodebin name=decoder ! queue ! video/x-raw ! videoconvert ! queue !
vp8enc deadline=1 ! rtpvp8pay ! queue ! whipsink name=ws use-link-
headers=true auth-token="hellothere" whip-endpoint="http://localhost:8080"
decoder. ! queue ! audio/x-raw ! opusenc ! rtpopuspay ! queue ! ws.
```

Fixes [gst-plugins-bad#1410 \(closed\)](#)

 GStreamer >  gst-plugins-rs > Merge requests > !949

Add a WebRTC WHEP source element

Code ▾

 Merged Sanchayan Maity requested to merge [SanchayanMaity/gst-plugins...](#) into [main](#) 3 months ago

All threads resolved! ⋮

Overview 153 Commits 16 Pipelines 37 Changes 7

This implements WHEP specification based on <https://datatracker.ietf.org/doc/html/draft-murillo-whep-00>

and has been tested with [Cloudflare](#).

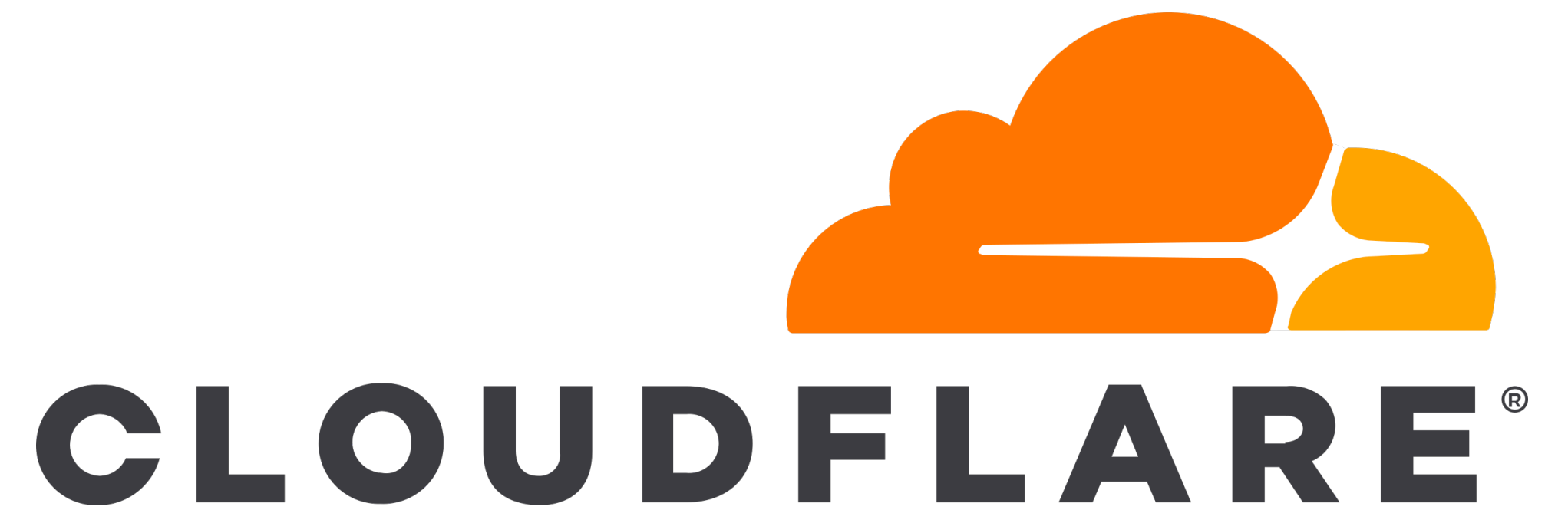
Closes [#237 \(closed\)](#).

Edited 3 months ago by Sanchayan Maity

 0  0  2

Platform support

 **Dolby.io**



So *yay.*

Using **WebRTC** for
ingress and **egress**
just got easier

Simulcast & SVC
are supported

It's **just** **HTTP**
transferring **SDP**

**Yup... SDP still
remains.**



But it does give
you freedom

Extra codecs?

OPUS RED?

RTP Header **Extensions**

DTX?

**It's not actually
groundbreaking at all**

(sorry Sergio)

**It's just HTTP
Offer Answer.**

OK there's some state handling too.

It gives everyone
2 common
protocols for
Send and Receive

Which leads
to...

Open

Innovation

Great Open Source Projects

SRT to WHIP

github.com/Eyevinn/srt-whip-gateway



SRT src -> WHIP sink
(yay 1.22 release)

WHEP Server

github.com/meetecho/simple-whep-server (Janus)

Are there others for other SFU/MCUs ?

WHEP Player

github.com/meetecho/simple-whep-client

github.com/Eyevinn/whep-video-component

WHIP Server

github.com/Eyevinn/whip

github.com/meetecho/simple-whip-server

WHIP Client

github.com/Eyevinn/whip

github.com/meetecho/simple-whip-client

github.com/ggarber/whip-go

**Its a great time to
start looking at
using WebRTC**

**Its a great time to
start looking at
using WebRTC**

Sidenote...

GStreamer

FTW.

GStreamer allows
you to pipe SRT,
WHIP, WHEP,
RIST, NDI...

And even

RTMP

**You don't
have to love
WebRTC.**

I certainly
don't love **it**
every day

But it is
incredibly
useful

And it is **another**
tool in the
toolbox for **sub**
second media

WHIP and **WHEP**

open up those

possibilities

Thanks!

OH!

**One more
thing...**


CommCon

2023 is

happening.

Announcements

imminent



COMMCON

commcon.xyz

dan@everycastlabs.uk
@dan_jenkins
@danjenkins@fosstodon.org

We're hiring (like everyone)

jobs.everycastlabs.uk

