Implementation Experience
WHIP in Janus (and GStreamer)

Lorenzo Miniero
lorenzo@meetecho.com

IETF 112, WISH WG
November 12th, 2021
Detailed blog post on integration

https://www.meetecho.com/blog/whip-janus-part-ii/
WISH-a-WHIP: WebRTC ingest for broadcasting

Lorenzo Miniero
@elminiero

IIT Real-Time Communication 2021 – WebRTC Track
October 13th 2021, Chicago, IL, USA

https://www.youtube.com/watch?v=b_QBd3WnGgY
A WHIP server based on Janus

• Janus is a popular WebRTC server, so a good option for WHIP
  • It implements its own JSON-based API, though (Janus API)
• Simple (and transparent) solution: basic API translator in front of Janus
  • WHIP API maps quite simply to set of Janus API primitives
  • No need to change anything in the WebRTC stack
• Implemented simple prototype using node.js and Express
  • REST server that implements the WHIP API, and talks to Janus accordingly
  • Only takes care of ingest: distribution out of scope

Simple WHIP Server
https://github.com/lminiero/simple-whip-server/
WHIP client and server + Janus

Simple WHIP Client

WHIP (HTTP)

Simple WHIP Server

Janus API

Janus

WebRTC
Mapping WHIP interactions to the Janus API

HTTP POST /endpoint (SDP Offer) -> validate token

attach handle (VideoRoom)
- success
- joinandconfigure (SDP Offer)
  - ack
  - event (joined, SDP Answer)

201 Accepted (SDP answer)
- address /resource from now on

WebRTC Producer -> WHIP Server

Janus

WebRTC Producer

WHIP Server

Janus
Mapping WHIP interactions to the Janus API

HTTP PATCH /resource (trickle)

find handle associated to /resource

trickle handle

success

204 No Content

ICE request

ICE response

DTLS setup

RTP flow
Mapping WHIP interactions to the Janus API

- WebRTC Producer
- WHIP Server
- Janus

HTTP PATCH /resource (new ufrag/pwd)
- new credentials --> ICE restart, generate SDP
- find handle associated to /resource

200 OK (new ufrag/pwd)

configure (SDP Offer, restart)
- ack
- event (configured, SDP Answer)

ICE request

ICE response
Mapping WHIP interactions to the Janus API

WebRTC Producer ➔ HTTP DELETE /resource ➔ cleanup endpoint ➔ find handle associated to /resource ➔ detach handle ➔ DTLS alert ➔ 200 OK ➔ success ➔ get rid of PeerConnection ➔ Janus

WebRTC Producer ➔ WHIP Server ➔ Janus
Mapping WHIP interactions to the Janus API

Diagram:
- WebRTC Producer
- WHIP Server
- Janus

Flow:
- DTLS alert
- PeerConnection is over
- event: hangup (DTLS alert)
- cleanup endpoint
- detach handle
- success
Writing a WHIP client for testing

• Needs to support HTTP (WHIP API) and have a WebRTC stack
  • Browsers are the obvious choice, but what about a native solution?
  • Many broadcasters today use custom tools (e.g., OBS)

• Unfortunately OBS-WebRTC is not currently an option
  • Used legacy WHIP API, and currently only supports Millicast ingestion

• Chose GStreamer’s **webrtcbin**\(^1\) for the purpose
  • Used it already with success in other applications (e.g., JamRTC)
  • Modular and very powerful, so easy to feed with external sources

---

**Simple WHIP Client**

[https://github.com/lminiero/simple-whip-client/](https://github.com/lminiero/simple-whip-client/)

---

\(^1\)[https://gstreamer.freedesktop.org/documentation/webrtc/]
WHIP client features (and limitations)

- Almost everything supported
  - Trickle (PATCH), STUN/TURN (via Link too), tokens, DELETE, etc.
  - Added support for non-trickle too (manual addition of candidates to SDP)
  - Option to force TURN (iceTransportPolicy: "relay" equivalent)
- Customizable audio/video pipelines
  - Easy to experiment with different sources and codecs
- A couple of things not supported in webrtcbin yet, though
  - ICE restarts (there seems to be a PR, though)
  - Link support in POST (we currently only do it via OPTIONS in the client)
Simple WHIP Client example

./whip-client -u https://mercury.conf.meetecho.com:8443/whip/endpoint/test \
-t hackathon \ 
-A "audiotests src is-live=true wave=red-noise ! audioconvert ! audioresample ! queue ! opusenc ! rtpopuspay pt=100 ! queue ! application/x-rtp,media=audio,encoding-name=OPUS,payload=100" \
-V "videotests src is-live=true pattern=ball ! videoconvert ! queue ! vp8enc deadline=1 ! rtpvp8pay pt=96 ! queue ! application/x-rtp,media=video,encoding-name=VP8,payload=96" \
-S stun.l.google.com:19302
Simple WHIP Client example

Plugin Demo: Video Room

Local Video

Remote Video #1 [WHIP Publisher 4321]

Remote Video #2

Remote Video #3

Remote Video #4

Remote Video #5

https://janus-legacy.conf.meetecho.com/videoroomtest.html?room=4321&subscriber-mode=true
Having fun with WHIP 😊

Having fun with WHIP 😊