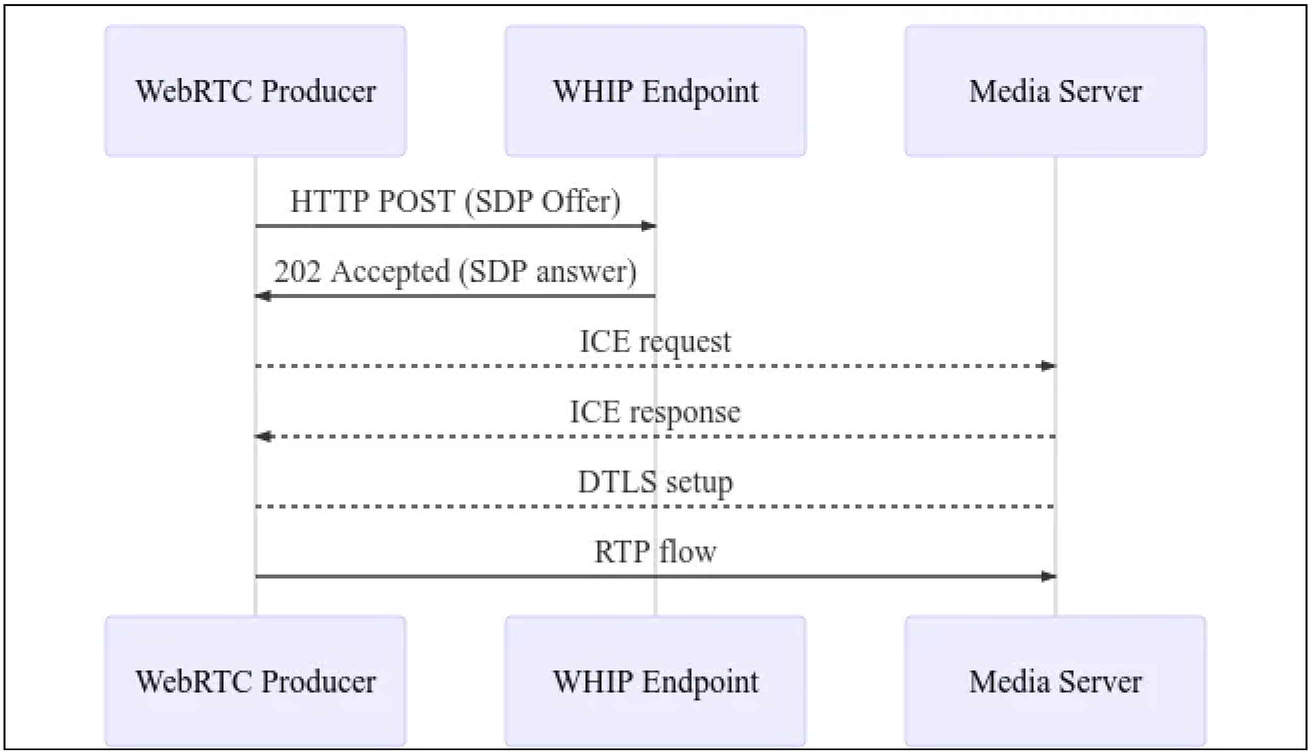
WebRTC publishing via WHIP

- Overview
    - Codecs supported
    - Operation flowchart
- Testing
- Authentication using Authorization: Bearer header
- Known issues

Overview

Since build 5.2.1816 WebRTC stream may be published via WebRTC-HTTP ingestion protocol (WHIP). The protocol proposes a standardized SDP exchange to establish a WebRTC connection using HTTP POST request



Today, WHIP is supported by OBS Studio 30 and later, but on Windows operating system only. WHIP stream publishing is almost the same as WebRTC publishing from browser except SDP exchange.

Codecs supported

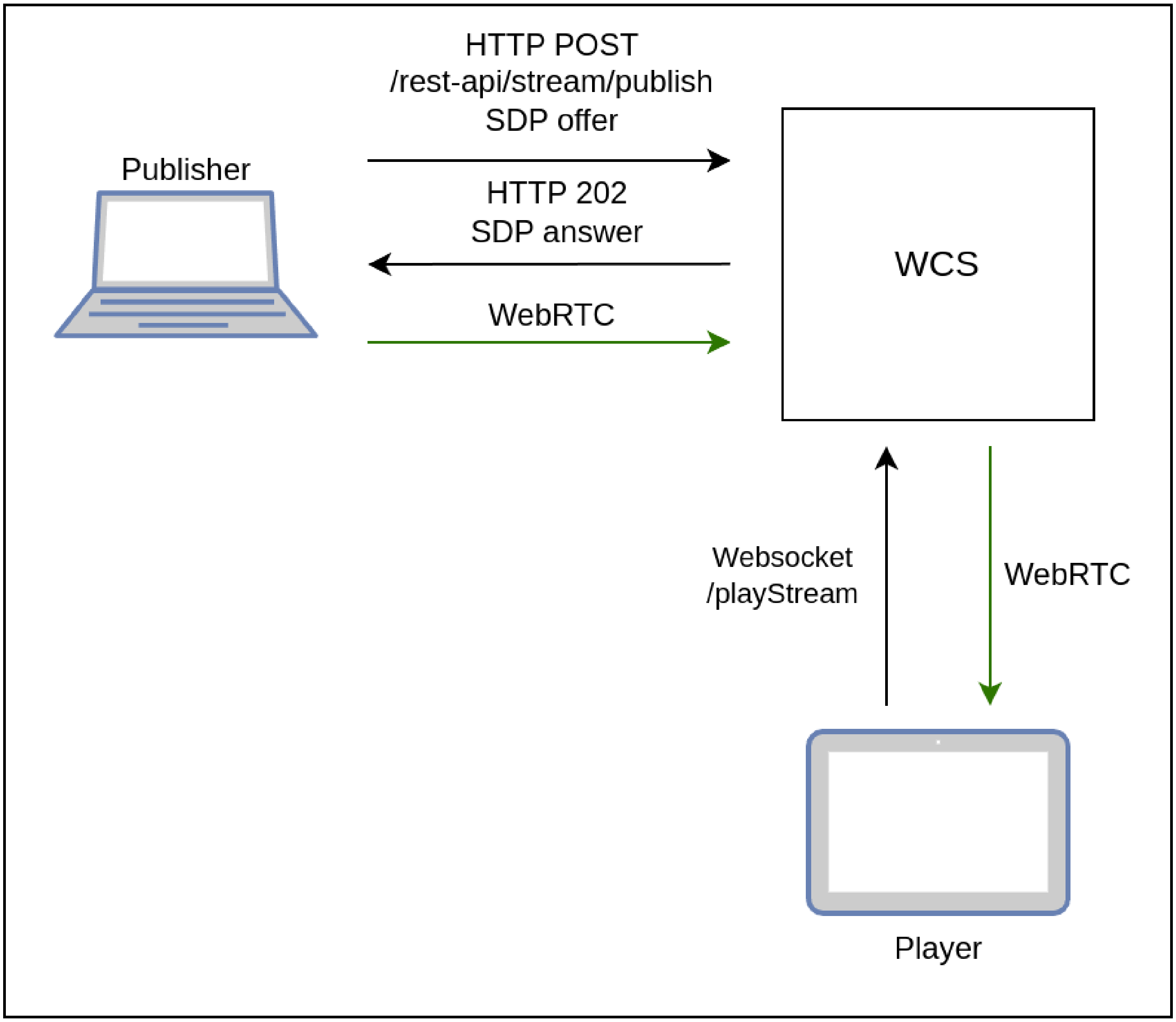
Video:

- H264
- VP8 if publisher supports it
- H265 if publisher supports it

Audio:

- Opus
- G711 (PCMA, PCMU) if publisher supports it

Operation flowchart



1. Publisher (OBS) sends HTTP POST query `/rest-api/stream/publish/streamName` with SDP offer

2. Publisher receives 202 Accepted with SDP answer

3. Publisher establishes WebRTC connection and publishes a stream

4. Player establishes Websocket connection and sends `playStream` command

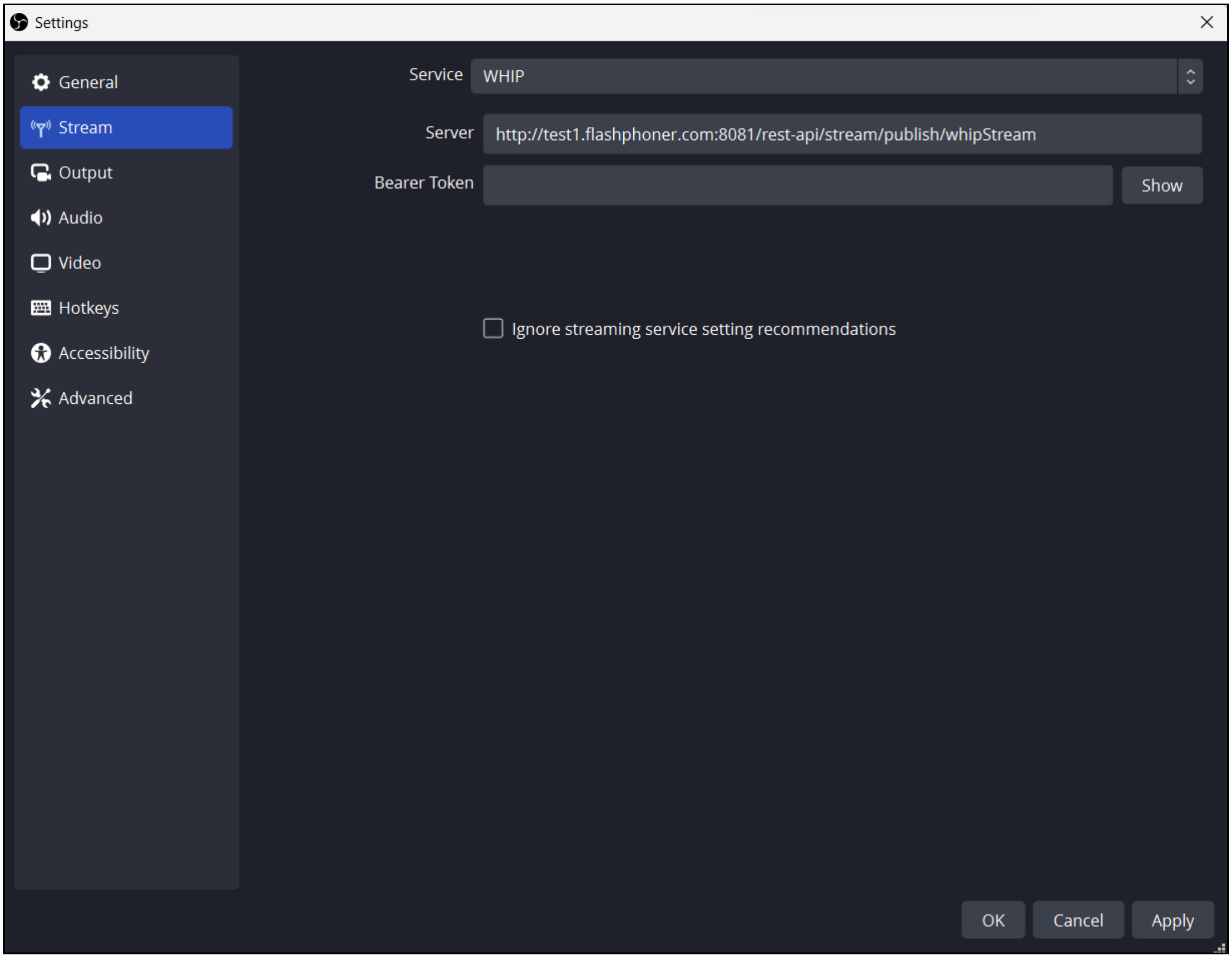
5. Player establishes WebRTC connection and receives the stream published

Testing

1. For test we use:

- WCS server
- OBS 30 and later for Windows to publish a stream
- Player example in Chrome browser to play the stream

2. On `Stream` tab in OBS settings choose `WHIP` service, enter HTTP POST query URL including a stream name to publish



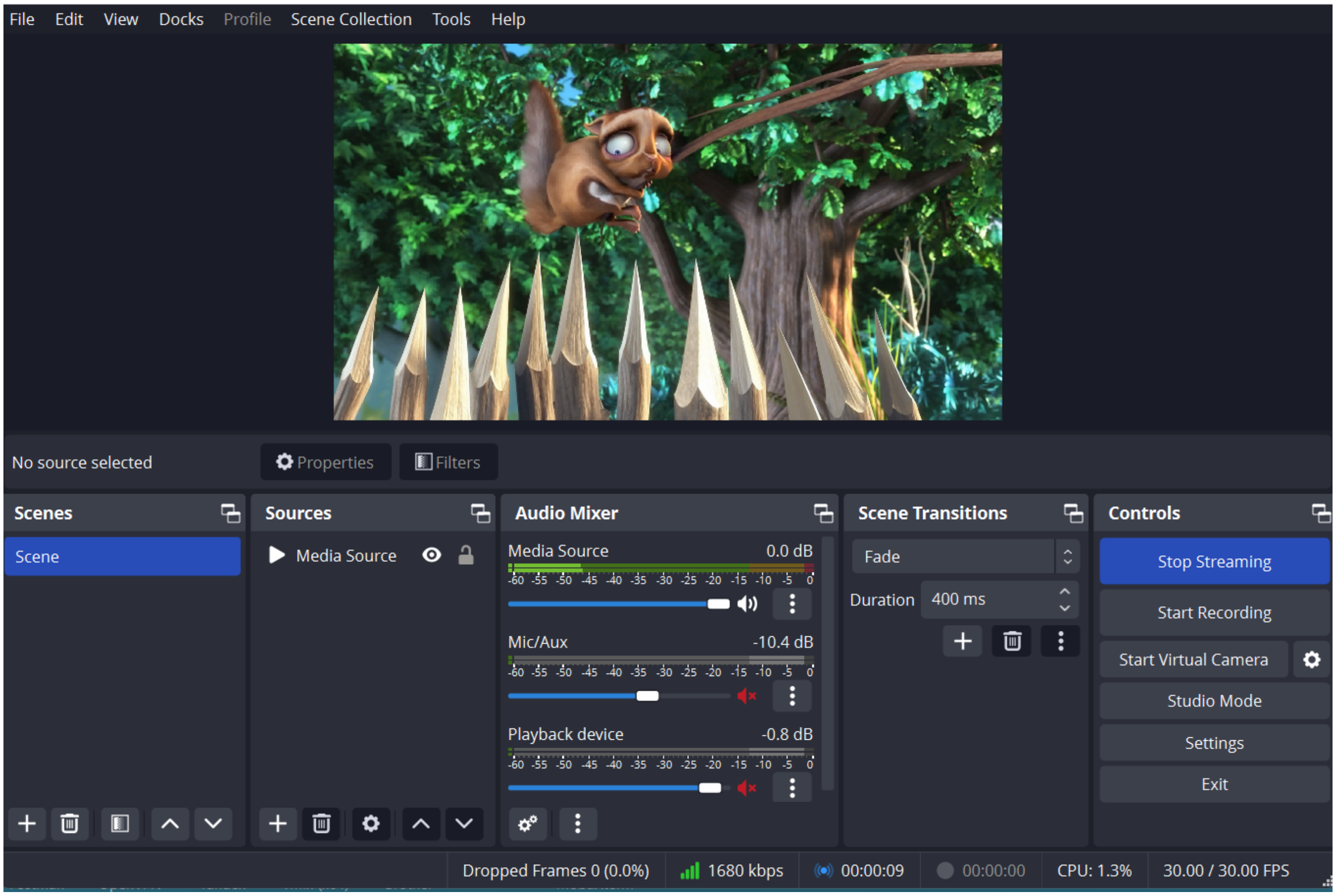
Service: WHIP

Server: http://test1.flashphoner.com:8081/rest-api/stream/publish/whipStream

Bearer Token:

□ Ignore streaming service setting recommendations

3. Publish the stream from OBS



No source selected

Scene

Media Source

Media Source 0.0 dB

Mic/Aux -10.4 dB

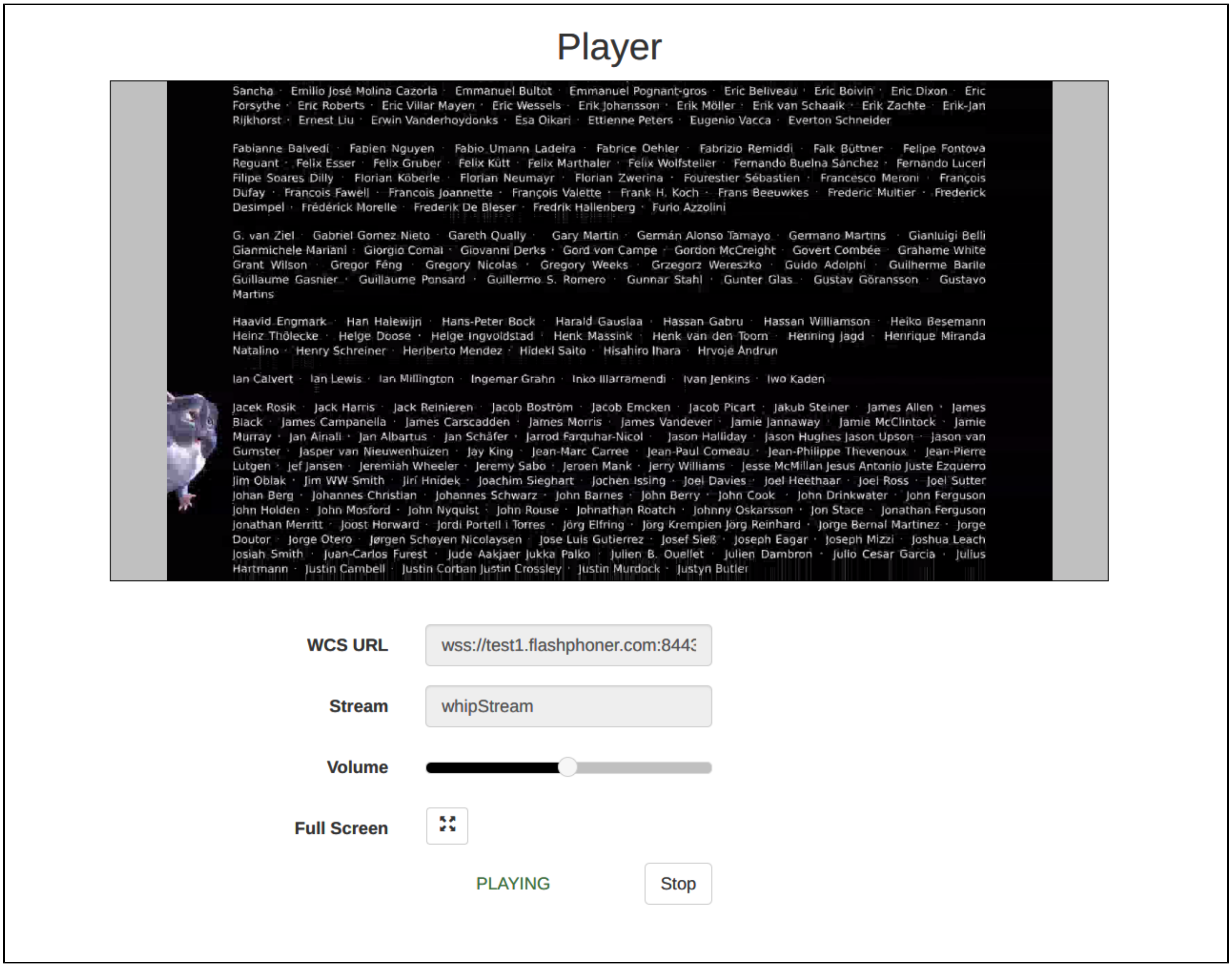
Playback device -0.8 dB

Fade

Duration 400 ms

Dropped Frames 0 (0.0%) | 1680 kbps | 00:00:09 | 00:00:00 | CPU: 1.3% | 30.00 / 30.00 FPS

4. Open Player example in Chrome browser from testing server, set the stream name to `whipStream` and click Play



Player

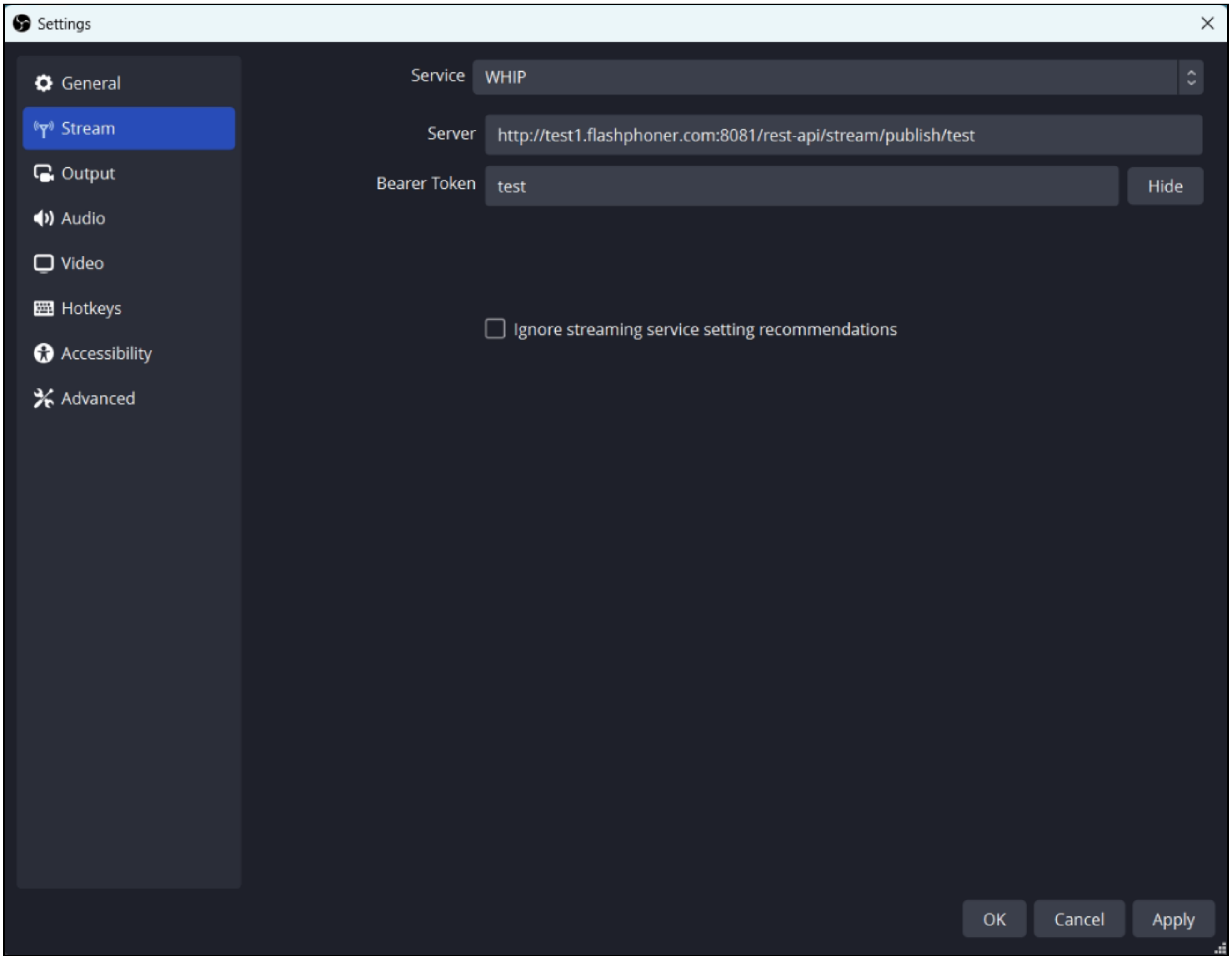
WCS URL: wss://test1.flashphoner.com:8443

Stream: whipStream

Sancha · Emilio José Molina Cazorla · Emmanuel Bultot · Emmanuel Pognant-gros · Eric Beliveau · Eric Boivin · Eric Dixon · Eric Forsythe · Eric Roberts · Eric Villar Mayen · Eric Wessels · Erik Johansson · Erik Möller · Erik van Schaaik · Erik Zachte · Erik-Jan Rijkhorst · Ernest Liu · Erwin Vanderhoydonks · Esa Oikari · Ettienne Peters · Eugenio Vacca · Everton Schneider
Fabianne Balvedi · Fabien Nguyen · Fabio Umann Ladeira · Fabrice Oehler · Fabrizio Remiddi · Falk Büttner · Felipe Fontova Reguant · Felix Esser · Felix Gruber · Felix Kütt · Felix Marthaler · Felix Wolfsteller · Fernando Buelna Sánchez · Fernando Luceri Filipe Soares Dilly · Florian Köberle · Florian Neumayr · Florian Zwerina · Fourestier Sébastien · Francesco Meroni · François Dufay · Francois Fawell · Francois Joannette · François Valette · Frank H. Koch · Frans Beeuwkes · Frederic Multier · Frederick Desimpel · Frédérick Morelle · Frederik De Bleser · Fredrik Hallenberg · Furio Azzolini
G. van Ziel · Gabriel Gomez Nieto · Gareth Qually · Gary Martin · Germán Alonso Tamayo · Germano Martins · Gianluigi Belli Gianmichele Mariani · Giorgio Comai · Giovanni Derks · Gord von Campe · Gordon McCreight · Govert Combée · Grahame White Grant Wilson · Gregor Fėng · Gregory Nicolas · Gregory Weeks · Grzegorz Wereszko · Guido Adolphi · Guilherme Barile Guillaume Gasnier · Guillaume Ponsard · Guillermo S. Romero · Gunnar Stahl · Gunter Glas · Gustav Göransson · Gustavo Martins
Haavid Engmark · Han Halewijn · Hans-Peter Bock · Harald Gauslaa · Hassan Gabru · Hassan Williamson · Heiko Besemann Heinz Thölecke · Helge Doose · Helge Ingvoldstad · Henk Massink · Henk van den Toorn · Henning Jagd · Henrique Miranda Natalino · Henry Schreiner · Heriberto Mendez · Hideki Saito · Hisahiro Ihara · Hrvoje Andrun
Ian Calvert · Ian Lewis · Ian Millington · Ingemar Grahn · Inko Illarramendi · Ivan Jenkins · Iwo Kaden
Jacek Rosik · Jack Harris · Jack Reinieren · Jacob Boström · Jacob Emcken · Jacob Picart · Jakub Steiner · James Allen · James Black · James Campanella · James Carscadden · James Morris · James Vandever · Jamie Jannaway · Jamie McClintock · Jamie Murray · Jan Ainali · Jan Albartus · Jan Schäfer · Jarrod Farquhar-Nicol · Jason Halliday · Jason Hughes Jason Upson · Jason van Gumster · Jasper van Nieuwenhuizen · Jay King · Jean-Marc Carree · Jean-Paul Comeau · Jean-Philippe Thevenoux · Jean-Pierre Lutgen · Jef Jansen · Jeremiah Wheeler · Jeremy Sabo · Jeroen Mank · Jerry Williams · Jesse McMillan Jesus Antonio Juste Ezquerro Jim Oblak · Jim WW Smith · Jiri Hnidek · Joachim Sieghart · Jochen Issing · Joel Davies · Joel Heethaar · Joel Ross · Joel Sutter Johan Berg · Johannes Christian · Johannes Schwarz · John Barnes · John Berry · John Cook · John Drinkwater · John Ferguson John Holden · John Mosford · John Nyquist · John Rouse · Johnathan Roatch · Johnny Oskarsson · Jon Stace · Jonathan Ferguson Jonathan Merritt · Joost Horward · Jordi Portell i Torres · Jörg Elfring · Jörg Krempien Jörg Reinhard · Jorge Bernal Martinez · Jorge Doutor · Jorge Otero · Jørgen Schøyen Nicolaysen · Jose Luis Gutierrez · Josef Sieß · Joseph Eagar · Joseph Mizzi · Joshua Leach Josiah Smith · Juan-Carlos Furest · Jude Aakjaer Jukka Palko · Julien B. Ouellet · Julien Dambron · Julio Cesar Garcia · Julius Hartmann · Justin Cambell · Justin Corban Justin Crossley · Justin Murdock · Justyn Butler
Volume
Full Screen
PLAYING
Stop

Authentication using Authorization: Bearer header

Since WCS build 5.2.1964 authentication using `Authorization: Bearer` header is supported



Service: WHIP

Server: http://test1.flashphoner.com:8081/rest-api/stream/publish/test

Bearer Token: test

☐ Ignore streaming service setting recommendations

In this case a publishing client (OBS) should send the header with the bearer token set with any query used when the stream is publishing via WHIP protocol. Any alphanumeric string may be used as token value, the stream name for example. The combination of the stream name and token value is checked at server side. If the token value differs from the value received with `/rest-api/stream/publish` query, the server returns `409 Permission denied`

Known issues

1. OBS does not support WebRTC publishing via TCP

Symptoms: a stream cannot be published from OBS when `ice_tcp_transport=true` parameter is used in server settings

Solution: switch to UDP at server side

```
ice_tcp_transport=false
```