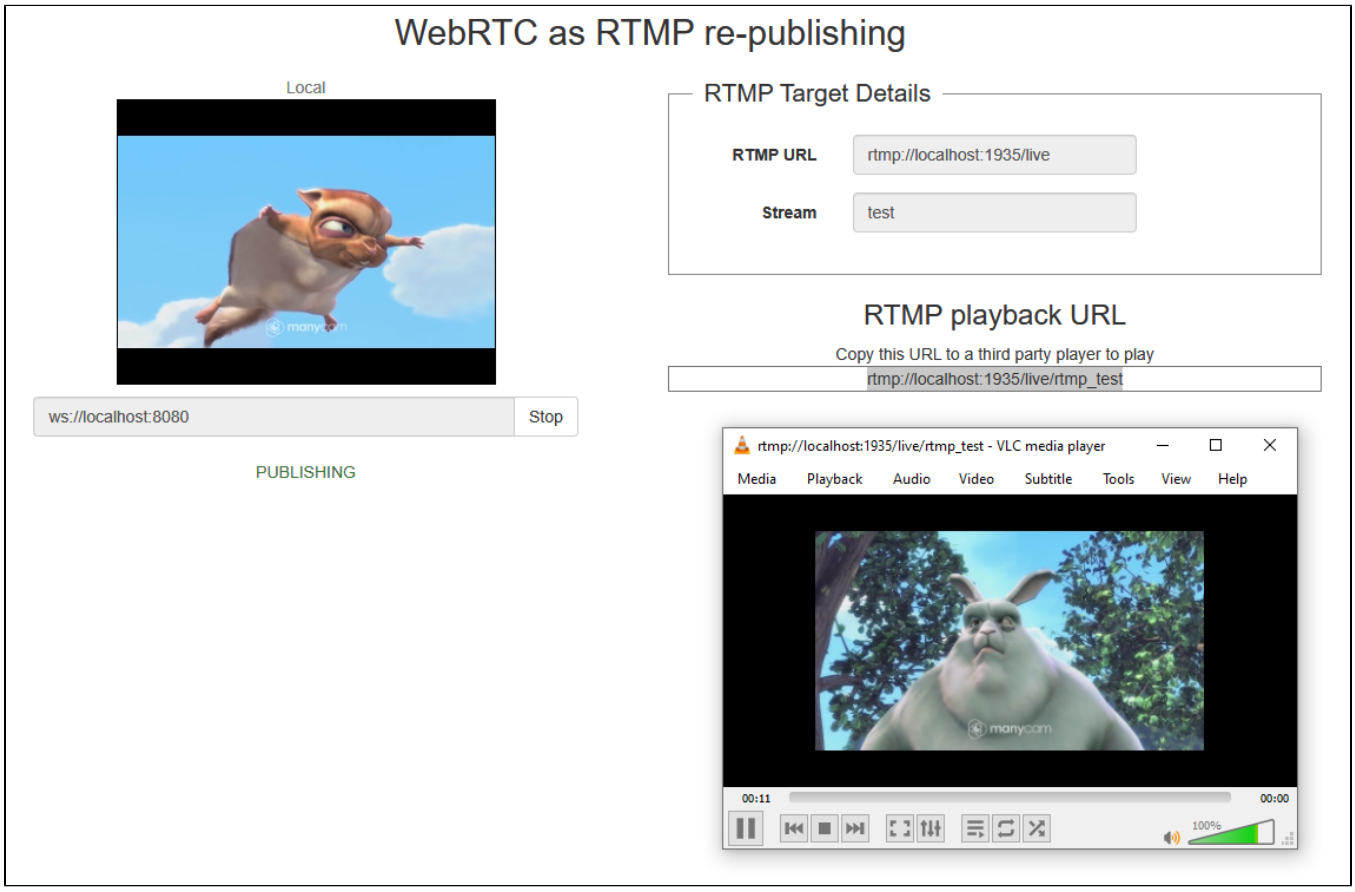
WebRTC as RTMP re-publishing

Example of republishing a WebRTC audio / video stream as RTMP

This example shows how you can send audio and video from the browser to the server with the WebRTC technology and redirect the received traffic to the same or another server via RTMP.

On the below screenshot, the browser has established connection to the WCS server and is sending audio and video to the server that performs republishing
to localhost via the RTMP protocol.



To play the redirected stream, copy the RTMP URL to external RTMP player, VLC for example

Code of the example

The source code of this example can be found on the WCS server at:

/usr/local/FlashphonerWebCallServer/client2/examples/demo/streaming/webrtc-as-rtmp-republishing

webrtc-as-rtmp-republishing.css – cascade style sheet file
webrtc-as-rtmp-republishing.html- example page
webrtc-as-rtmp-republishing.js -script

You can test this example at this address:

https://host:8888/client2/examples/demo/streaming/conference/webrtc-as-rtmp-republishing.html

Here, host is the address of the WCS server.

Analyzing the code

To examine the code, let's take the version of thewebrtc-as-rtmp-republishing.jsfile with the hash of ecbadc3, which is available here and can be downloaded with the corresponding build 2.0.212.

The script establishes connection to the WCS server and manages publishing of the WebRTC stream. In addition, the script places the test RTMP player on the webpage and passes the resulting RTMP address to it for playback.

1. Initialize API

Flashphoner.init() code

```
Flashphoner.init();
```

2. Connect to the server.

Flashphoner.createSession() code

```
        Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
            ...
        });
```

3.Receiving the event confirming successful connection

ConnectionStatusEvent ESTABLISHED code

```
        Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
            //session connected, start streaming
            startStreaming(session);
        }).on(SESSION_STATUS.DISCONNECTED, function(){
            ...
        }).on(SESSION_STATUS.FAILED, function(){
            ...
        });
```

3. Send WebRTC video stream with republishing to RTMP server.

session.createStream(), stream.publish() code

When the createStream() method is used to create the stream, aside from the standard parameters this field is also specified rtmpUrl. It contains the RTMP address of the server this stream will be republished to. The name of the republished RTMP stream is rtmp_{streamName}, where rtmp_ is the standard prefix set in the flashphoner.properties file.

For example, if streamName=stream1, then the resulting RTMP stream will be named rtmp_stream1

```
    session.createStream({
        name: streamName,
        display: localVideo,
        cacheLocalResources: true,
        receiveVideo: false,
        receiveAudio: false,
        rtmpUrl: rtmpUrl
        ...
    }).publish();
```

5.Receiving the event confirming successful streaming

StreamStatusEvent PUBLISHING code

```
    session.createStream({
        ...
    }).on(STREAM_STATUS.PUBLISHING, function(publishStream){
        setStatus(STREAM_STATUS.PUBLISHING);
        onStarted(publishStream);
        sendDataToPlayer();
    }).on(STREAM_STATUS.UNPUBLISHED, function(){
        ...
    }).on(STREAM_STATUS.FAILED, function(){
        ...
    }).publish();
```

6. Forming the RTMP URL to display on the page and copy to an external player

sendDataToPlayer() code

```
function sendDataToPlayer() {
    var player = document.getElementById("player");
    var host = field("rtmpUrl")
        .replace("localhost", window.location.hostname)
        .replace("127.0.0.1", window.location.hostname);

    var rtmpStreamPrefix = "rtmp_";
    var url = host + "/" + rtmpStreamPrefix + field("streamName");
    player.setURLtoFlash(url);
}
```