FFmpeg

Please [read how to find the setup a match (/scoring/create-match/sources/third-party)](/scoring/create-match/sources/third-party) first.

This is a software developer concept. MAS does not provide additional support around this.

As FFmpeg natively supports RTMP, you can follow the official streaming guide (http://trac.ffmpeg.org/wiki/StreamingGuide), read up on RTMP options (http://underpop.online.fr/f/ffmpeg/help/rtmp.htm.gz) and set the match RTMP ingest destination (/scoring/create-match/sources/third-party#rtmp) as the output destination for FFmpeg.

You will want to take note of FFmpeg best practices here, don't encode at a resolution/framerate beyond what the network/device is capable of doing. This command seems to work well on most modern devices where some sort of graphics chip is available (i.e. you shouldn't *need* a standalone GPU):

```
ffmpeg -y
-re -f {input_format}
-analyzeduration 10M -probesize 10M -i {input_sources}
-c:v libx264 -preset ultrafast -tune zerolatency
-b:v 6000k -maxrate 6000k -bufsize 12000k
-g {frame_rate * 2}
-f flv {rtmp_destination}
```

You'll want to play with this a bit - consider in particular 30FPS/1080P maximum input on the server and toggling the bitrate up/down to trade performance/network/image-quality (resize and delay/drop as necessary).

You can read more about this here:

http://trac.ffmpeg.org/wiki/EncodingForStreamingSites (http://trac.ffmpeg.org/wiki/EncodingForStreamingSites)

Websocket Frame Data

This is an advanced software developer concept. MAS does not provide additional support around this.

If you're looking to develop alongside MAS, you may be interested in the websocket frame data receiver (this is how MAS is able to do its video streaming out of the web browser).

MediaRecorder

The output of the web API MediaRecorder (https://developer.mozilla.org/en-US/docs/Web/API/MediaRecorder) can be relayed to the websocket ingest URL directly.

Codecs

The Websocket ingest, like RTMP, is restricted to H.264 but will re-encode audio inputs. This has been tested on latest versions of:

✅ Chrome/Edge desktop (and other probably most other Chromium forks)

✅ Safari desktop

✅ Safari iOS/iPadOS

And has been tested to *not work* on:

❌ Android web (only produces VP9)

❌ Firefox (only produces VP8)

There may be some unexpected variation on the above if your device does not have H.264 encoding options, such as some devices sold in the European Union and Asia.

FFmpeg Pipe

You can direct the output of FFmpeg to a pipe and use a websocket to perform the actual streaming - especially useful if you don't like RTMP, need better metrics, or want to use more advanced codecs.

Ensure you are encoding H.264/AAC and containerise it as FLV (or any other stream friendly format (i.e. don't use a container that relies on special byte instructions like the MOOV atom in MP4)).

This is largely redundant at the moment, but as we begin to support more modern video codecs this will become more useful.