FIDAL Open Calls

Ektacom StreamSelector

User Guide



|  |  |
| --- | --- |
| Delivery | FIDAL Open calls StreamSelector user guide |
| Project Reference | FIDAL |
| Package |  |
| Target | FIDAL, EKTACOM |
| Main Recipy |  |
| Delivery Date | 07/27/2024 |
| Status | Version 1 |

Contributors

|  |  |  |
| --- | --- | --- |
| Main authors | | |
| Julien COURTAT | EKTACOM | V1 : 07/27/2024 |
|  |  |  |
|  |  |  |
|  |  |  |
| Co-authors | | |
|  |  |  |

Summary

This document describes Ektacom Network Application StreamSelector. This application is a video gateway operating in a Kubernetes cluster in the special configuration of FIDAL Open Calls. It process high quality video stream in real time with low latency in order to maximize the end user experience. This document only applies to deployments in one of the FIDAL testbeds at Telenor, University of Patras and University of Malaga.

This document also briefly explains standard protocols involved in the streaming solution and how a partner application can interact with the StreamSelector to accelerate innovation in the world of live video streaming.

Table des matières

[1 Video streaming in the context of the FIDAL project 5](#_Toc172727121)

[1.1 Real Time Protocol (RTP) 5](#_Toc172727122)

[2 Ektacom Stream selector Presentation 8](#_Toc172727123)

[2.1 software architecture 8](#_Toc172727124)

[2.2 Streaming features 8](#_Toc172727125)

[2.2.1 Accessing live stream session 8](#_Toc172727126)

[2.2.2 Aggregating live stream sessions 10](#_Toc172727127)

[3 StreamSelector workflow 13](#_Toc172727128)

[3.1 Accessing live stream session 13](#_Toc172727129)

[3.1.1 Contact Ektacom 13](#_Toc172727130)

[3.1.2 Request creation of a session 14](#_Toc172727131)

[3.1.3 Streaming video 14](#_Toc172727132)

[3.1.4 Accessing video Stream 15](#_Toc172727133)

[3.2 Aggregating live stream sessions 17](#_Toc172727134)

[3.2.1 Contact Ektacom 17](#_Toc172727135)

[3.2.2 Request creation of two sessions and start streaming 17](#_Toc172727136)

[Stream aggregation : create StreamSelector mosaic using webUI 18](#_Toc172727137)

[Stream aggregation : create StreamSelector mosaic using REST api 21](#_Toc172727138)

[3.2.3 Accessing aggregated stream 21](#_Toc172727139)

[3.2.4 Stopping stream aggregation 22](#_Toc172727140)

[4 Annexes 23](#_Toc172727141)

[4.1 Annex A 23](#_Toc172727142)

[4.2 Annex B 23](#_Toc172727143)

[4.3 Annex C 23](#_Toc172727144)

[4.4 Annex C 24](#_Toc172727145)

|  |  |
| --- | --- |
| **Abbreviation** | **Definition** |
| **ABR** | Adaptive Bitrate |
| **CPU** | Central Processing Unit |
| **FPS** | Frames Per Second |
| **(U)HD** | (Ultra) High Definition |
| **HDR** | High Dynamic Range |
| **HLS** | HTTP Live Streaming |
| **HTTP** | Hypertext Transfer Protocol |
| **IETF** | Internet Engineering Task Force |
| **JSON** | JavaScript Object Notation |
| **KLV** | Key-Length-Value |
| **RTCP** | Real-time Transport Control Protocol |
| **RTP** | Real Time Protocol |
| **SSRC** | Synchronization Source |
| **TCP** | Transmission Control Protocol |
| **UDP** | User Datagram Protocol |
| **URL** | Unified Resource Locator |

# Video streaming in the context of the FIDAL project

The video streaming chain is in charge of collecting video contributions, that come from devices of professionals or users, and distributing them to end users, to be viewed by using standard applications or dedicated one developed in the context of FIDAL open calls. These viewing applications will use data formats and protocols, following international standards or *de-facto* open standards.

Video captured in the field at the events, will have to be transferred to the FIDAL system using data formats and protocols, following international standards, or *de-facto* open standards, enabling the implementation of required technologies to handle them.

The following requirements apply to the contribution of video for the FIDAL use cases:

* Only use formats defined in international standards or *de-facto* open standards, for content coding, multiplexing, storing and the transport layer.
* To have latency compatible with glass-to-glass applications.
* Support up to UHD (4K resolution) formats.
* To be compatible with smartphones when used as capturing devices.

At the same time, full interoperability must be guaranteed for a variety of different users, having different video rendering capabilities and network contexts. Encoding tools therefore need to be optimised for allowing adaptive streaming while producing the right video representation in an efficient and low complexity manner.

The Ektacom StreamSelector component receives video streams coming from both professionals and users (UGC), with the support of several streaming formats. This video content can be forwarded to the StreamSelector component that has the role of transforming the video for the purpose of each Fidal project features.

## Real Time Protocol (RTP)

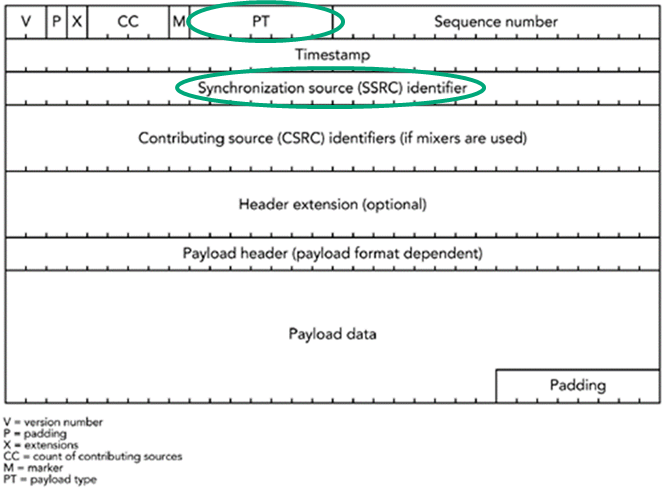
Video contribution defines the delivery of video content to video production, *i.e.*, to professionals that will oversee editing, transcoding, or any other necessary operation for later delivering the content to the audience. In the context of FIDAL video contribution corresponds to the streaming of video from User Devices or Professional equipment to the StreamSelector component (a video gateway), which is the entry point to FIDAL streaming chain.

RTP (IETF RFC3550) can be used without additional protocols, with dedicated applications that manage connections between the UGC provider and the server. RTP is more than 20 years old, but still a relevant and simple means to transport media data over IP. It can satisfy low latency applications and can also be augmented by several compatible protocols in order to ensure a more reliable or more secure transmission, or session management protocols. It continues to be updated, for example with the recent work at Internet Engineering Task Force (IETF) for defining the RTP payload format for the latest MPEG video codec standard, MPEG-I VVC[[1]](#footnote-1). Given a suitable configuration of the video codec, it is possible to achieve ultra-low latency streaming with RTP, with as low as sub-frame duration latency. RTP can therefore satisfy streaming for various latency and network requirements, while being robust towards packet loss. RTP supports the latest video codec technologies such as MPEG-H HEVC or even the new MPEG-I VVC.

All video streams related by this document rely on RTP to transport video across devices.

When a partner wants to stream video to the platform, its application asks permission from the StreamSelector register (Ektacom Estendirect api), where its right to transmit is verified. The technical information needed for streaming is then sent back to the application (IP and port number of the streaming destination, the RTP SSRC unique session value). The partner application is then able to stream to the StreamSelector Edge component using these parameters.

The RTP format complies to Multiplexed RTP specifications as described in RFC 8860[[2]](#footnote-2) “Sending Multiple Types of Media in a Single RTP Session”. For more details see also RFC 8872[[3]](#footnote-3) “Guidelines for Using the Multiplexing Features of RTP to Support Multiple Media Streams”. In a RTP multiplexed stream all media data of that stream uses the same SSRC field of RTP packet headers, and can be differentiated by using their Payload Type (PT) value which has to be different, as illustrated in the next figure.



RTP packet header structure with location of SSRC stream identification information

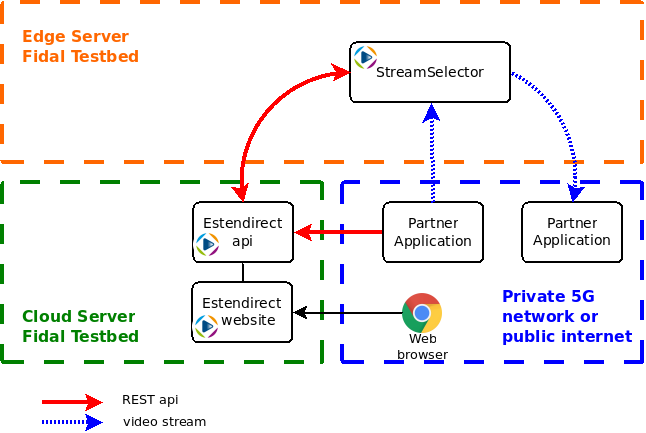
Implementers of applications that will use StreamSelector have to pay attention to setup expected SSRC value in their streams, and write appropriate PT values, aiming at differentiating media data types. How to get the SSRC value to use is explained in detail later in this document.

StreamSelector supports three media types: video, audio, and Key-Length-Value (KLV, SMPTE ST 336[[4]](#footnote-4)) metadata. The metadata contains information about the device, like location or type of device. The video is encoded using MPEG-H HEVC and the audio is encoded using MPEG-4 AAC. Video resolution can go up to the resolution of the smartphone camera (nowadays, UHD resolution, a.k.a 4K).

Media data included into RTP streams can be:

* Video: MPEG-H HEVC/ITU-T H.265 Main Profile, progressive video
* Audio: MPEG-4 AAC
* Metadata: SMPTE ST 336/KLV UAS local data set

# Ektacom Stream selector Presentation



Video stream characteristics:

* encoded with HEVC, max resolution 2160p50
* transport protocol is RTP

## software architecture

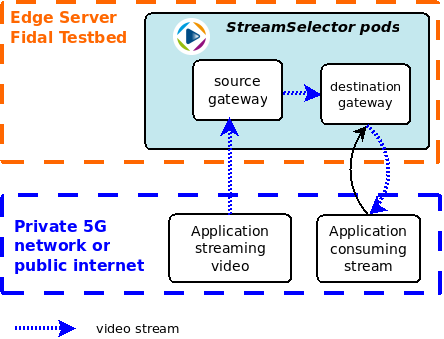
The FIDAL platform targets the possibility of being executed in a private or public Cloud environment. The FIDAL streaming chain has been defined accordingly and the following choices have been made:

* The components and sub-components of the streaming chain operate as containers, within Kubernetes Pods.
* They can run in real-time on usual Cloud hardware architectures.
* They are highly portable and compatible with Linux-based operating systems.
* RTP is not a protocol that Cloud platform load balances with specific equipments, meaning all trafic will enter the Kubernetes cluster with NodePort service type at the cost of being tunneled to the node that deploy StreamSelector source gateway

## Streaming features

### Accessing live stream session

This is the most basic feature proposed by the StreamSelector. It consists in forwarding video streams being lively received by the platform. It represents a unique access point for stream both streamers and consumers, simplifying video stream delivery workflow.



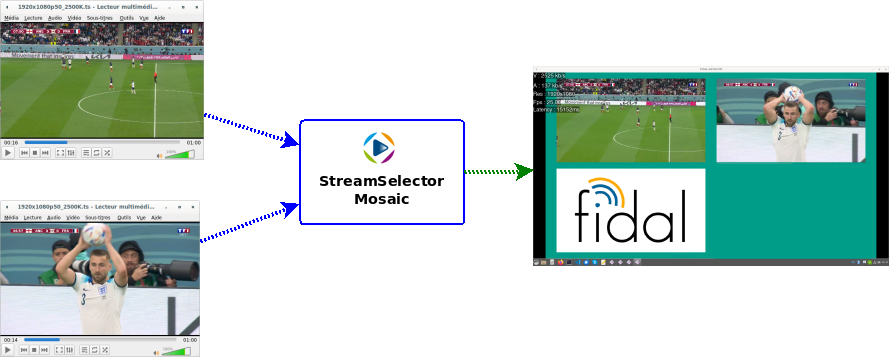
The stream identifier is the SSRC field of the RTP messages received by the StreamSelector source gateway. According to the value of that field, the gateway forwards the message to another gateway which in turn will wait for a stream consuming application to request for that stream on same SSRC.

Maximum number of parallel video streaming depends on each testbed deployment (number of source/destination gateway pods).

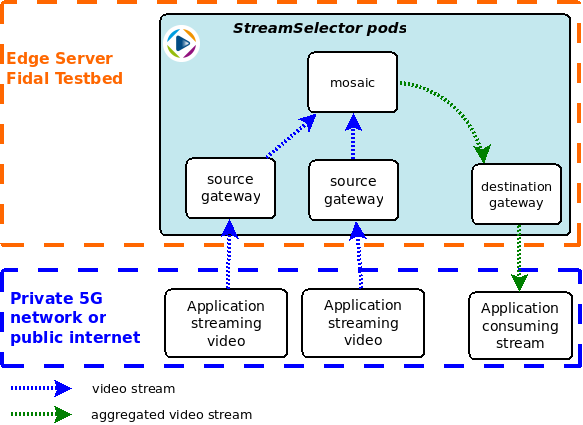
An example of video streaming command is provided in Annex B of this document.

### Aggregating live stream sessions

This is enhanced feature proposed by the StreamSelector. It consists in aggregating a maximum of 2 live video streams using the StreamSelector mosaic application. Aggregating video streams means displaying streams of choice into a new one made with a mosaic layout. It represents a unique access point for stream both streamers and consumers, simplifying video stream delivery workflow.



This is the high-level architecture of the StreamSelector for video stream aggregation:



Maximum number of parallel video aggregation depends on each testbed deployment.

# StreamSelector workflow

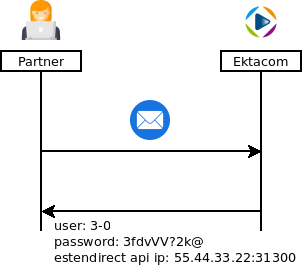
In order to take advantage of the features proposed by Ektacom StreamSelector network application, partners shall apply to the following workflow.

## Accessing live stream session

StreamSelector is not open to any external application. Open call partner must first register with FIDAL representative then contact Ektacom or the manager in charge of the StreamSelector deployment within targeted testbed. Note that credentials provided for establishing access to live session may not be the same as ones provided for stream aggregation. Take care to connect with credentials corresponding to what your application expects.

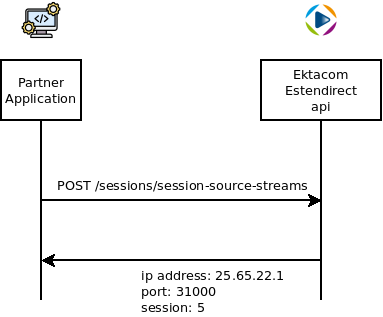
### Contact Ektacom

Send an email to Ektacom ([fidal.opencalls@ektacom.com](mailto:fidal.opencalls@ektacom.com)) which will create accounts and all required stuff on the Estendirect platform according to partner needs. Ektacom will provide credentials required for partner software when connecting to StreamSelector.



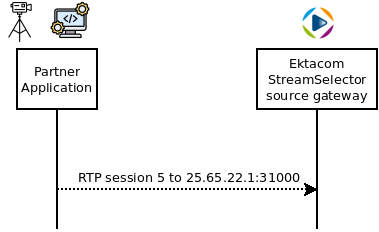
### Request creation of a session

Before starting video streaming, partner application must request creation of a new session at /sessions/session-source-streams of Estendirect platform api described in the Annex A of this document. The platform responses the IP address or FQDN of one of the StreamSelector gateway that will handle the stream, as well as a RTP session id.



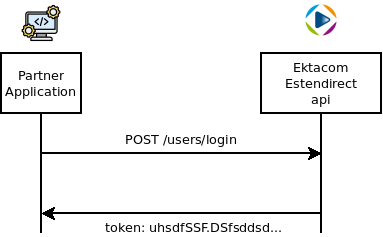
### Streaming video

Partner application can now send video using RTP to the IP or FQDN provided on step 2. Take care to not pause video streaming for more than 30 seconds, because StreamSelector would consider the stream as stopped and release cluster resources to the profit of new sessions.



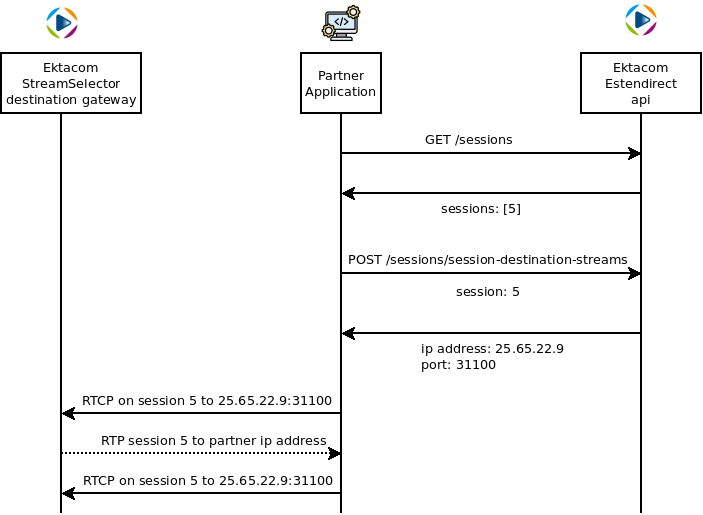
### Accessing video Stream

The video live stream can now be accessed by any application. This application must log in to /users/login of the Estendirect api, get a token back to present in next api requests.



Partner application would list available sessions using /sessions api, then request access to that session at /sessions/session-destination-streams api.

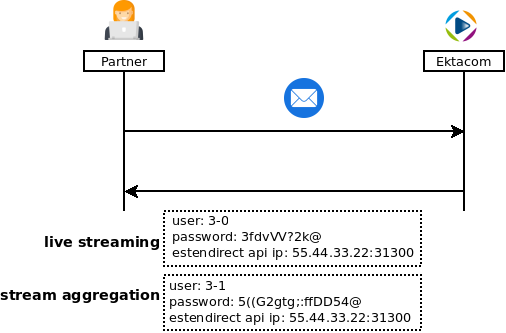
In order to receive video stream, partner application must send a RTCP message with selected session to the IP address and port returned at previous step. This will trigger the StreamSelector gateway to start sending stream to the source IP of the RTCP message. In order to maintain streaming, partner application must periodically send RTCP message with same session id (every 10 seconds) to the same IP address, else the StreamSelector would consider the application as disconnected and release resources.



## Aggregating live stream sessions

### Contact Ektacom

Send an email to Ektacom ([fidal.opencalls@ektacom.com](mailto:fidal.opencalls@ektacom.com)) which will create accounts and all required stuff on the Estendirect platform according to partner needs. Ektacom will provide credentials required for partner software when connecting to StreamSelector.



Partner shall receive two different kinds of credentials:

1. live streaming, same as ones received in 3.1.1
2. stream aggregation, allowing to connect to Estendirect webUI to create mosaic application that is capable of aggregating streams.

### Request creation of two sessions and start streaming

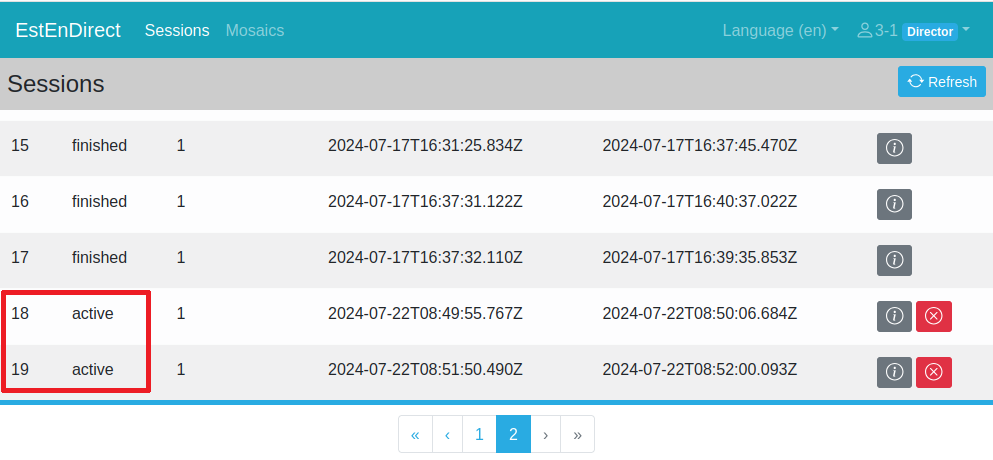
Before starting video streaming, partner application must request creation of two new sessions at /sessions/session-source-streams. Follow instructions of 3.1.2 and 3.1.3 to create sessions and stream them to StreamSelector.

### Stream aggregation: create StreamSelector mosaic using webUI

Connect to Estendirect webUI using stream aggregation credentials

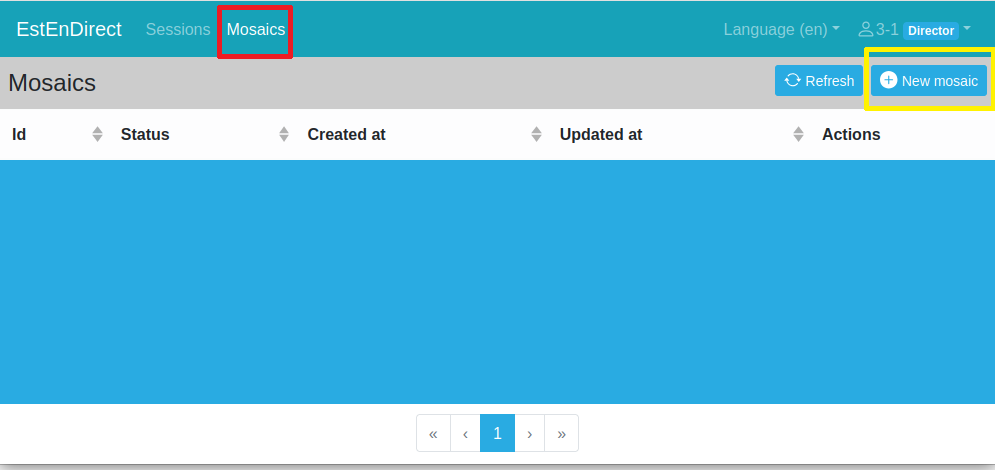


Make sure live stream sessions are alive

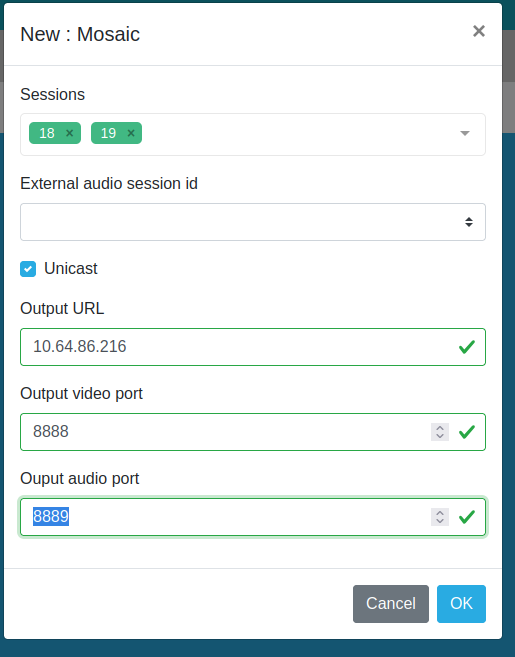


If just created sessions are not marked as active, there is no chance for the StreamSelector mosaic to aggregate them. If one or many sessions are marked as finished, restart procedure stated in 3.1.2 and 3.1.3.

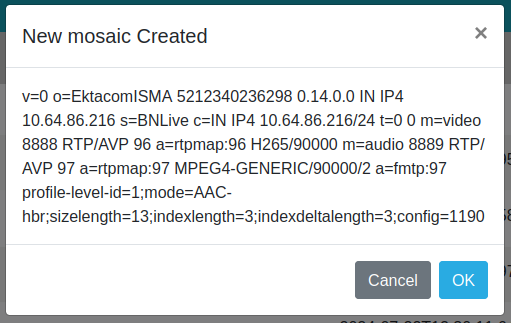
If sessions are active, go to mosaic creation frame (red square, then yellow one).



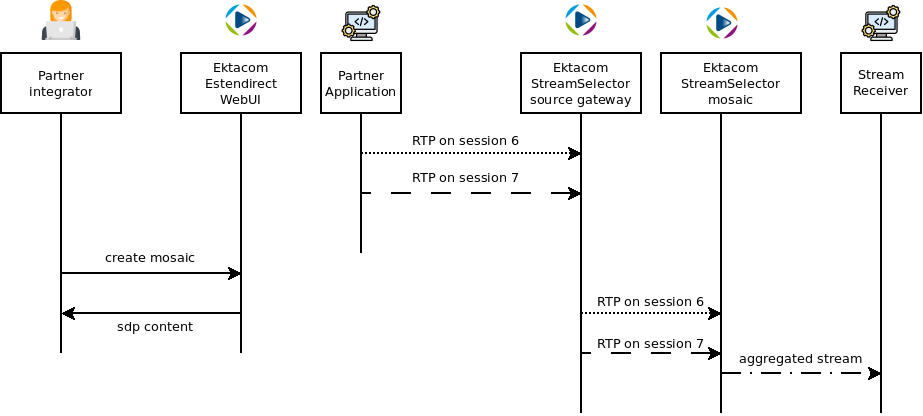
Fill mosaic frame parameters by selecting the sessions of choice. Make sure to tick Unicast checkbox and fill the Ouput URL testbox with the ip address or FQDN of the host which will receive the aggregated stream. Fill Output video port as well as Ouput audio port textboxes because RTP protocol splits video and audio streams.



On mosaic creation, the webUI returns the content of a SDP file that can be provided to an external tool to playback the video stream.

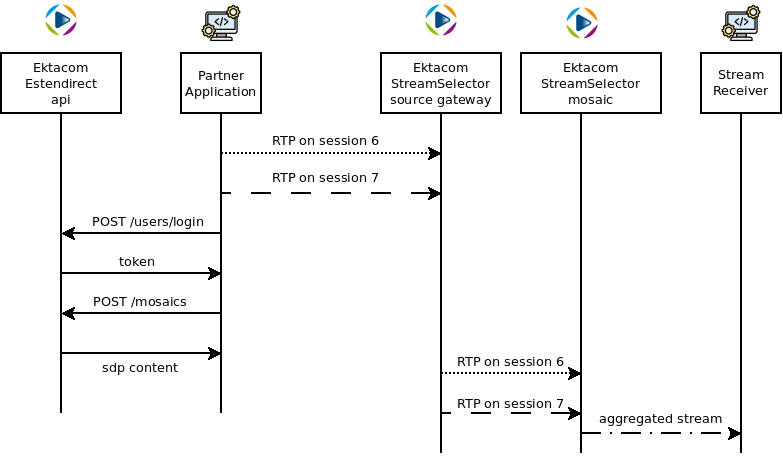


The following time chart resumes all these steps, it is supposed that RTP sessions have been created as stated in 3.1.2.



### Stream aggregation: create StreamSelector mosaic using REST api

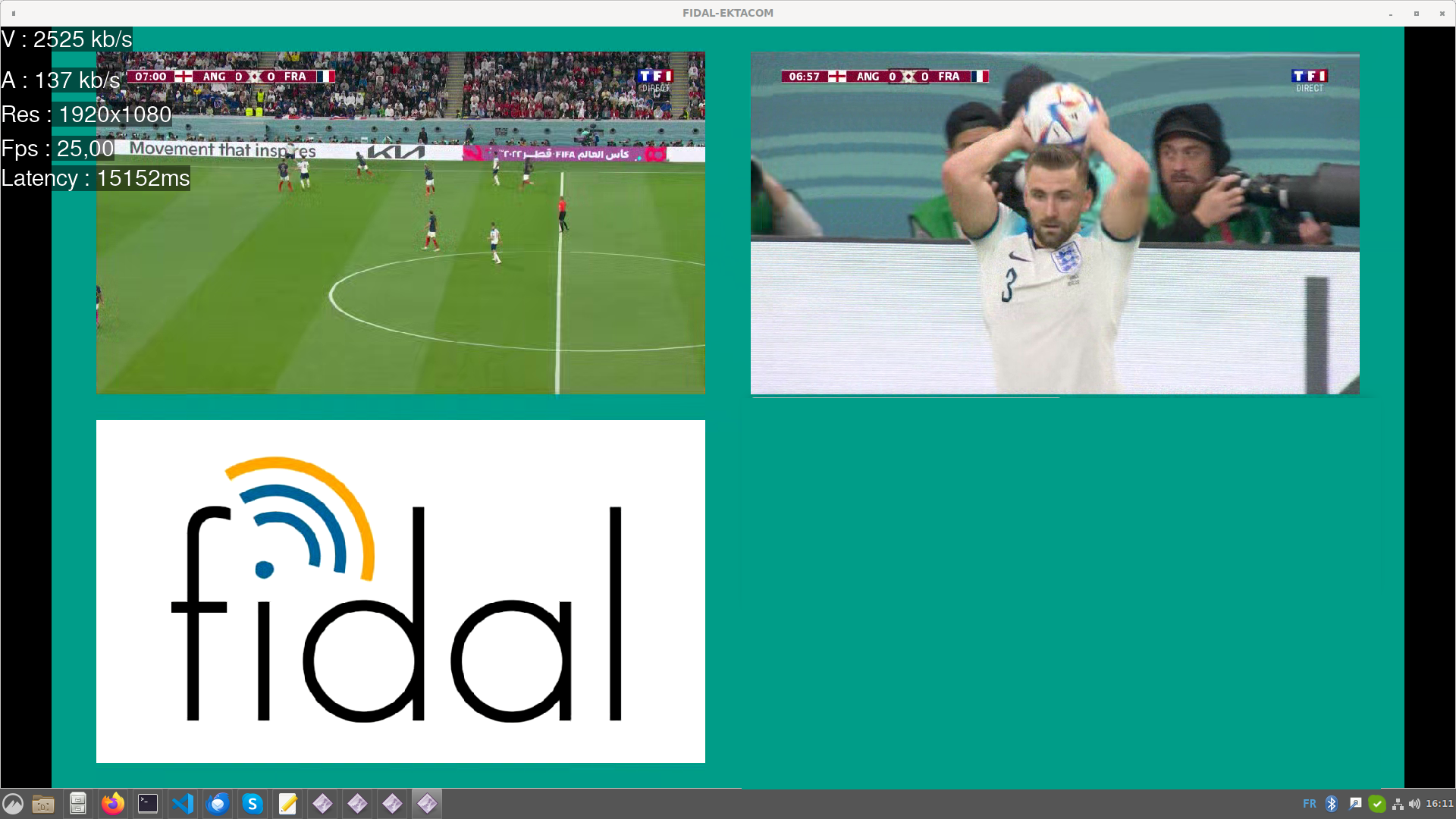
Once familiar with mosaic creation using the webUI, partner application can automate processing with StreamSelector REST api. To do the same as previous chapter, send POST request to /mosaics resource.



### Accessing aggregated stream

External tools such as VLC or ffplay can parse this sdp to play the aggregated streams.

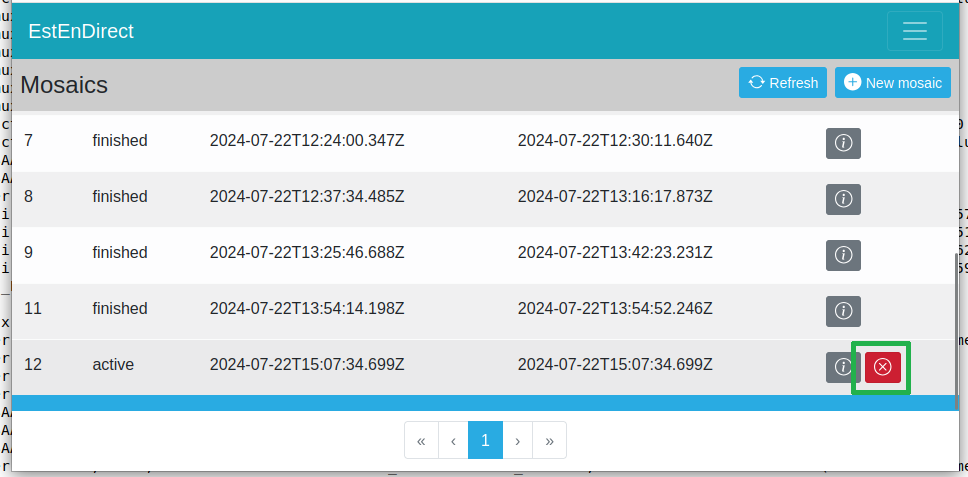
|  |
| --- |
| ffplay -protocol\_whitelist file,udp,rtp,sdp -i /tmp/mosaic.sdp |



### Stopping stream aggregation

Live Stream aggregation is server resources consuming, because it rescales input streams then encodes the output using HEVC codec on the fly. Nowadays video resolution such as 1920x1080p requires the StreamSelector to reserve 8 vCPU to aggregate things. Based on that knowledge, number of deployed mosaics per cluster is not infinite. Only one mosaic instance is reserved to each partner.

Partner must release the mosaic once input streams has stopped in order to be able to create a new one.



Release mosaic can also be done using REST api on /mosaics/$id/stop

# Annexes

## Annex A

StreamSelector REST api subset for Fidal OpenCall partners is provided as an external html document (streamselector-fidal-opencalls-rest-api-2.0.0.0.zip).

## Annex B

Example of terminal command that sends video stream to a host

|  |
| --- |
| ffmpeg -stream\_loop -1 -re -i 1920x1080p50\_2500K.ts -map 0:0 -c:v copy -an -ssrc 18 -f rtp 'udp://1.2.3.4:30001?pkt\_size=1316' |

## Annex C

|  |  |  |
| --- | --- | --- |
| **Title** | **Protocols/Standard** | **Details** |
| HTTP Live Streaming | Apple HLS  IETF RFC 8216 | <https://tools.ietf.org/html/rfc8216> |
| Real Time Messaging Protocol | Adobe RTMP | <https://rtmp.veriskope.com/pdf/rtmp_specification_1.0.pdf> |
| Real Time Protocol (RTP) | IETF RFC3550 | <https://tools.ietf.org/pdf/rfc3550.pdf> |
| RTP Payload Format for H.264 Video | IETF RFC 6184 | <https://tools.ietf.org/pdf/rfc6184.pdf> |
| RTP Payload Format for High Efficiency Video Coding (HEVC) | IETF RFC 7798 | <https://tools.ietf.org/pdf/rfc7798.pdf> |
| RTP Payload Format for Versatile Video Coding (VVC) | IETF Draft | <https://tools.ietf.org/pdf/draft-ietf-avtcore-rtp-vvc-06.pdf> |
| RTP Payload Format for MPEG-4 Streams | MPEG-4 AAC RTP streaming format  IETF RFC 3640 | <https://tools.ietf.org/html/rfc3640> |
| Advanced Video Coding | MPEG-4 AVC/ITU-T H.264 | <https://www.iso.org/standard/75400.html> |
| High Efficiency Video Coding | MPEG-H HEVC/ITU-T H.265 | <https://www.iso.org/standard/75484.html> |
| Advanced Audio Codec | MPEG-4 AAC | <https://www.iso.org/standard/36083.html> |

## Annex C

Credits:

icons by flaticon.com/auteurs/uniconlabs

End of Document

1. <https://www.iso.org/obp/ui/#iso:std:iso-iec:23090:-3:ed-2:v1:en> [↑](#footnote-ref-1)
2. <https://www.rfc-editor.org/rfc/rfc8860.pdf> [↑](#footnote-ref-2)
3. <https://www.rfc-editor.org/rfc/rfc8872.pdf> [↑](#footnote-ref-3)
4. <https://ieeexplore.ieee.org/document/8019807> [↑](#footnote-ref-4)