RDK-V Use Cases

All streaming will be done by the Media Streamer in the Gateway (Hybrid) device. The RMF source element in the Client box will receive the data and do the processing.

Live Streaming to Client Device

Here all the data is streamed from the media streamer through Home networking sink. HNSrc in the Client will receive the data from HNSink and playback using the RMF pipeline.
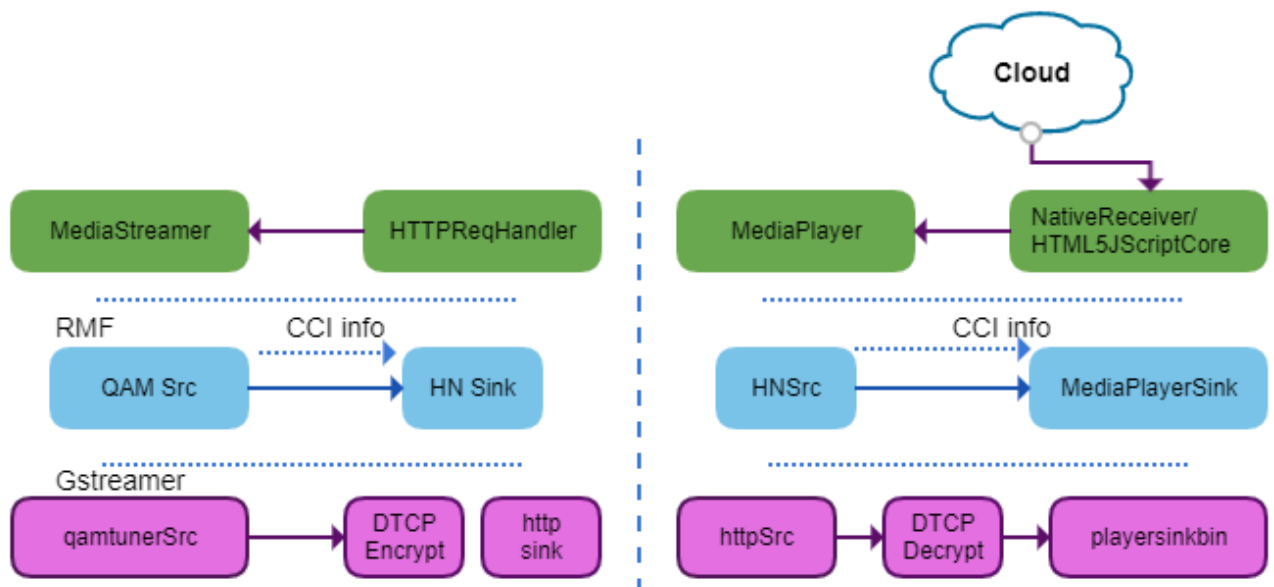
- Here SPTS (Single Program Transport Stream) data is streamed from the media streamer through Home networking sink. HN Src in the Client device will receive the data from HN sink and playback using the RMF pipe line.
- After receiving the request URI, the media server creates an RMFQAM source element and HN Sink element.
- RMFQAM source tunes to a particular program, starts pumping clear SPTS stream.
- Media Server connects the QAM Source to the HN Sink. HN Source receives the media at the client side for play out.
- In the case of encrypted media play out, the stream data is decrypted using the POD manager which is invoked by the QAM source. QAM Source sends the clear SPTS to the HN Sink.
- Media server registers to get events from QAM Source. Through the event mechanism QAM Source will send user authorization related information to the Media Server.
- The Media Server sends this information to client device.
- DTCP encryption and decryption are used for protecting contents over network.
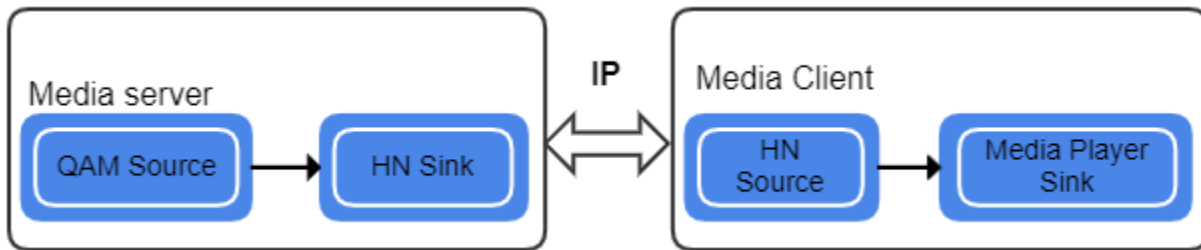
Live Streaming to Client Device : Use Case

- The right side of the figure is considered for client boxes.
- The left side of the figure is considered for gateway box.
- In case of live streamthe HTTPReqHandler will receive a request from client box and the MediaStreamer will set a pipeline between QAM Source and HN Sink.
- QAM Srcelement receives the data from the tuner and sends to the HN Sink.
- In the client side another pipeline has created between HNSrcand MediaPlayerSink.
- HNSrc receives the incoming data from HNSinkand feeds to the MediaPlayerSinkfor live playback.
- DTCP encryption and decryption are used for protecting contents over network.

The Block diagram for the live streaming is as follows:

Live Streaming to Client Device : Seq Diagram

Live streaming

Client   **MediaServer**   **QAMSrc**   **HNSink**

Client → MediaServer: GET(url)

> http://<serverip>:8080/vldms/tuner?
> ocap_locator=ocap://0x(srcId)

MediaServer → Client: HTTP response 200 OK

MediaServer → QAMSrc: getQAMSourceInstance(srcId)

> Create a new QAMSrc Instance if does
> not exist for the requested srcId. Get
> FPQ for the given srcId and tune.Listen
> for PSI changes.Set Media filters.
> Configure cypher and start decrypting if
> required.Get CCI values to notify
> connected sinks.

QAMSrc → MediaServer: qamSrc

MediaServer → QAMSrc: setEvents(QAMSrcEvent)

> Register for QAMSrc error events:
> Tuner not locked.No PAT,No PMT,no
> authorization,unrecoverable error end
> note MediaServer->+HNSink : HNSink()
> note left of HNSink
> DTCP enabled/disabled.
> socket descriptor.
> chunked mode transfer

HNSink → MediaServer: HNSink

MediaServer → HNSink: init()

MediaServer → HNSink: setSINKProperties(DTCP =0, socketId, Chunked=1)

MediaServer → HNSink: setEvents(HNSinkEvent)

> Register for HNSink error events:
> Socket error.

MediaServer → HNSink: setSource(qamSrc)

MediaServer → QAMSrc: play(time_pos=0, speed=1.00)

HNSink → Client: Stream SPTS

Client   **MediaServer**   **QAMSrc**   **HNSink**