Building a Media Pipeline

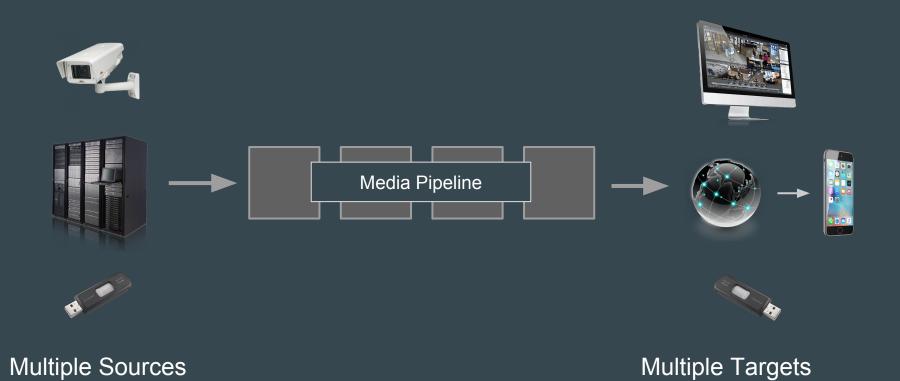
•••

Tom Distler

2019

https://github.com/tjdistler/docs

What Problem Are We Trying to Solve?



Video Renderer

RTP Source (network)

Playback Source (network)

> File Source

Data Transforms

Audio Renderer

RTP Server (network)

MPEG-2 TS Server (network)

File Sink

RTP Source (network)

Playback Source (network)

> File Source

H.264 Decoder H.264 Encoder

MPEG4 Decoder G.711 Decoder

G.728 Decoder Video Scaler

AAC Decoder AAC Encoder Video Renderer

Audio Renderer

RTP Server (network)

MPEG-2 TS Server (network)

File Sink

RTP Source (network)

Playback Source (network)

File Source

Video Metadata

Audio

Metadata

Source

Selector

*Data **Normalizers**

H.264 Decoder

MPEG4 Decoder

G.728 Decoder

AAC Decoder

AAC Encoder

H.264

Encoder

G.711

Decoder

Video

Scaler

Video Renderer

Re-timer

Splitter

Muxer

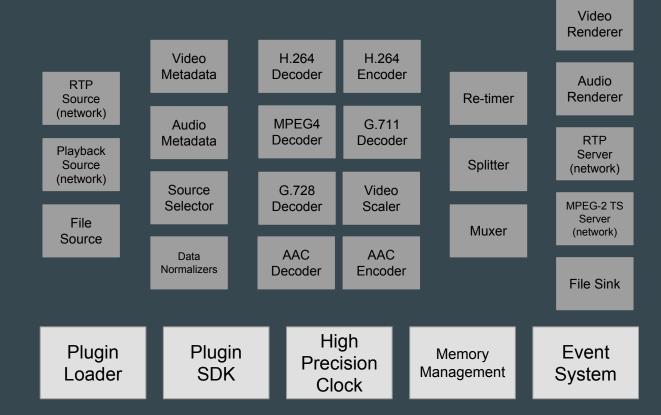
Audio Renderer

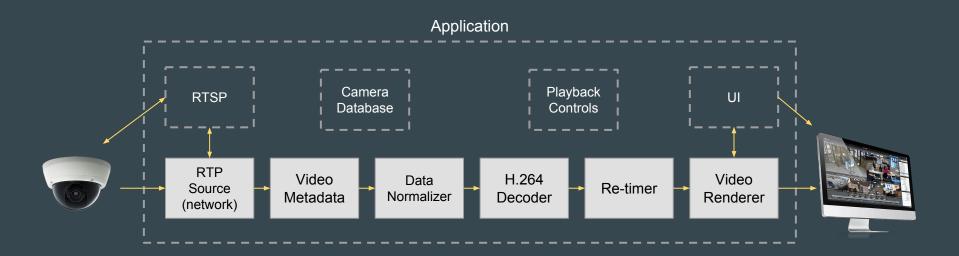
RTP Server (network)

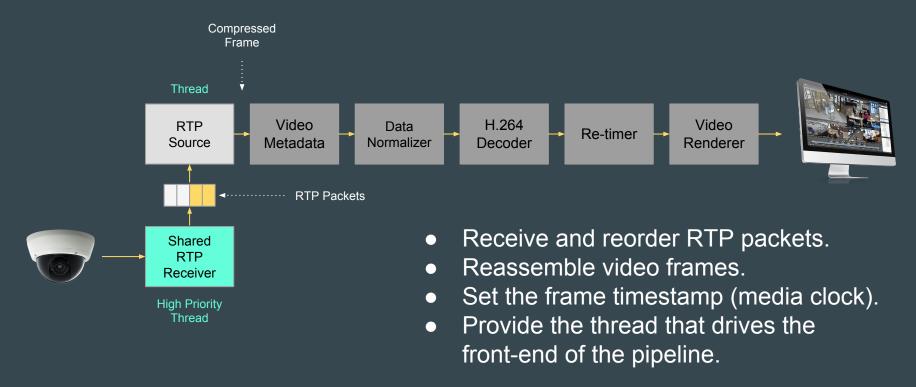
MPEG-2 TS Server (network)

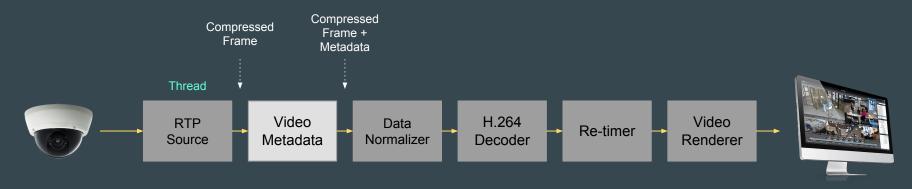
File Sink

*camera make/model specific

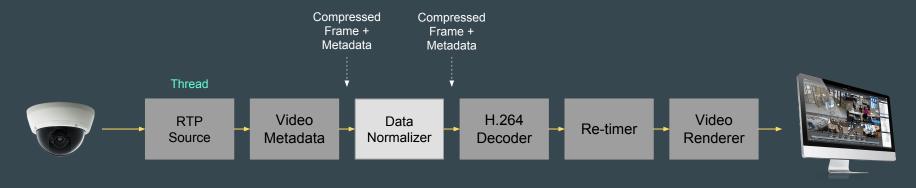




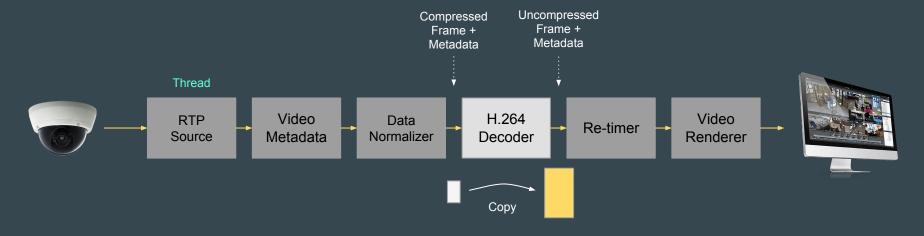




- Determine video codec.
- Read media-info headers for width/height, etc.
- Attach metadata and UTC timestamp to buffer.

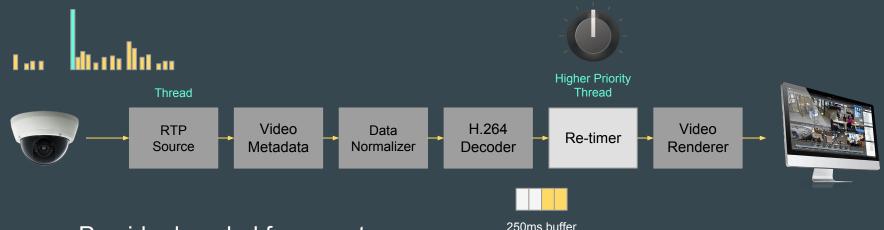


- Correct camera make/model specific problems:
 - Incorrect RTP clock frequency.
 - Missing/incorrect frame dimension information.

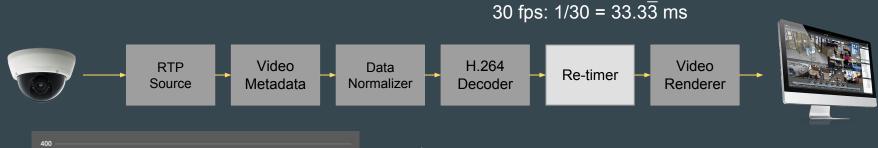


- Decodes into fixed size buffers.
- Decode times:
 - 18-28 ms for I-frames
 - 7-14 ms for P-frames

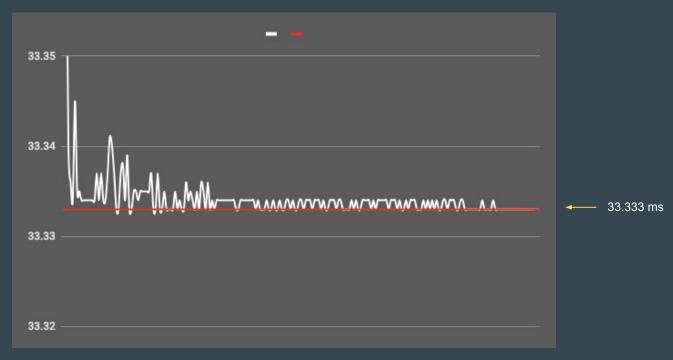




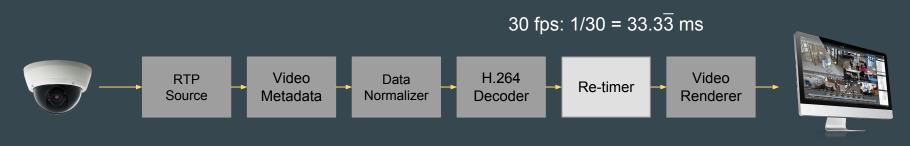
- Provide decoded frames at a consistent rate to the renderer.
- Correct for input stream jitter.
- Handle low-latency vs buffered mode.
- Buffer frames during playback.



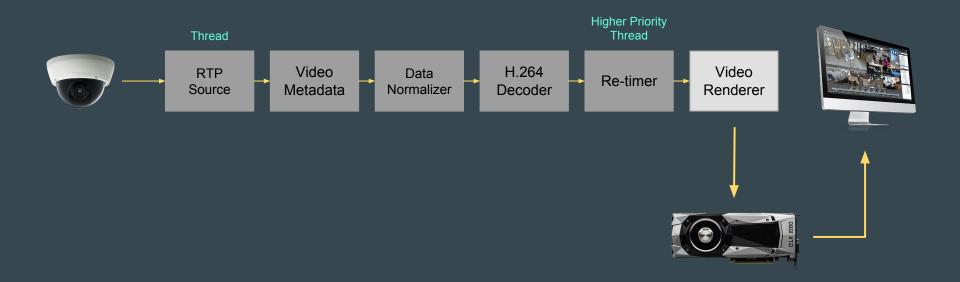
```
now = read_clock()
slept = now - prev
prev = now
sleep_time -= slept // 33.33 - 34.5 = -1.17
sleep_time += target // -1.17 + 33.33 = 32.16
sleep((int)sleep_time) // 32 ms
```

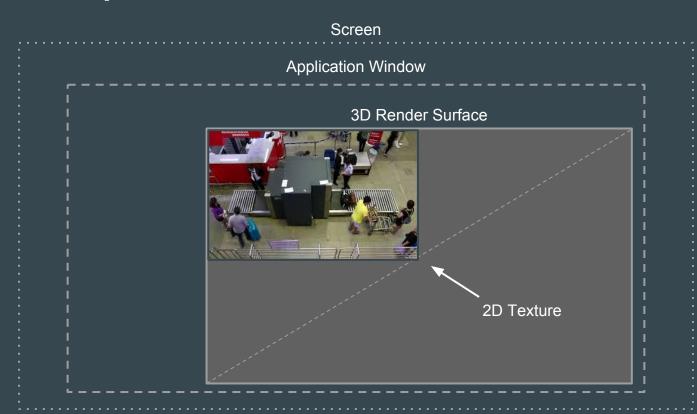


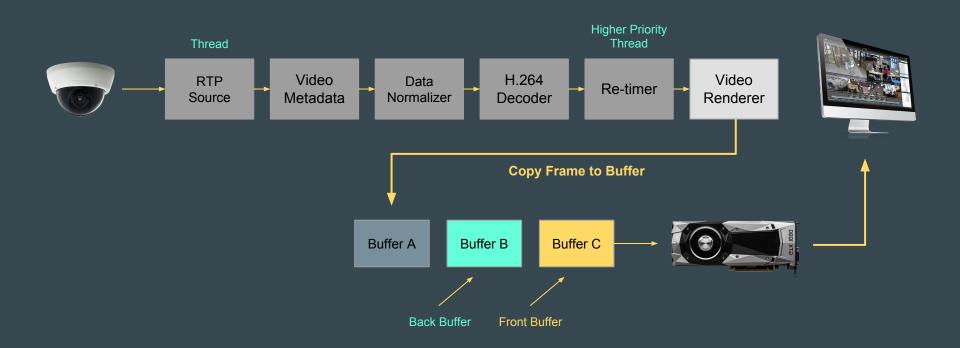
Average frame timing over 5 minutes

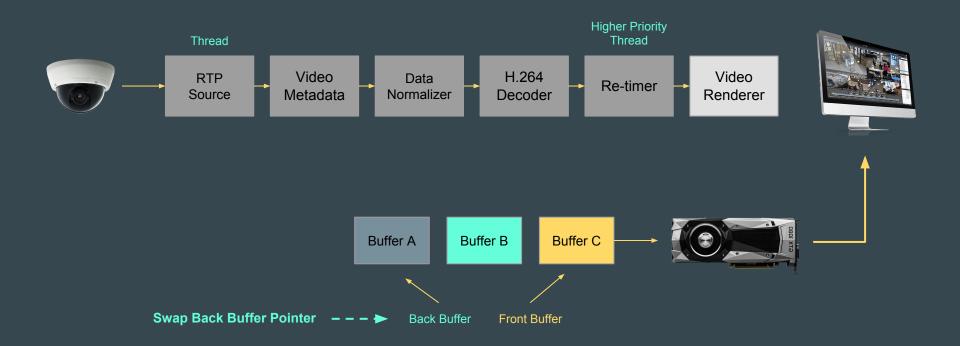


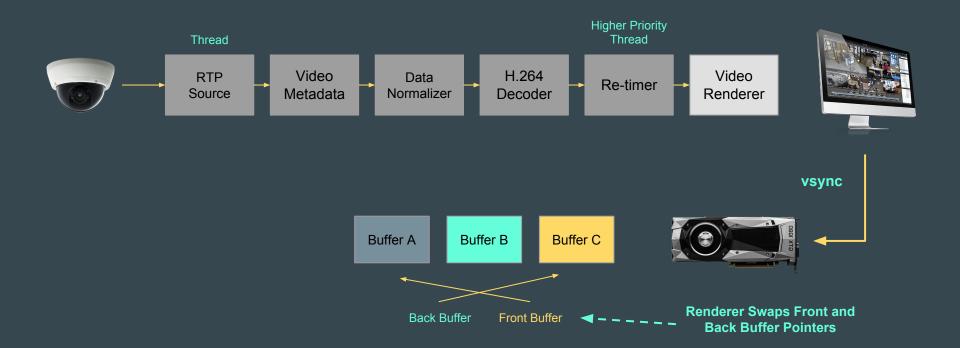
```
now = read_clock()
slept = now - prev
prev = now
sleep_time -= slept
sleep_time += target
sleep((int)sleep_time)
```

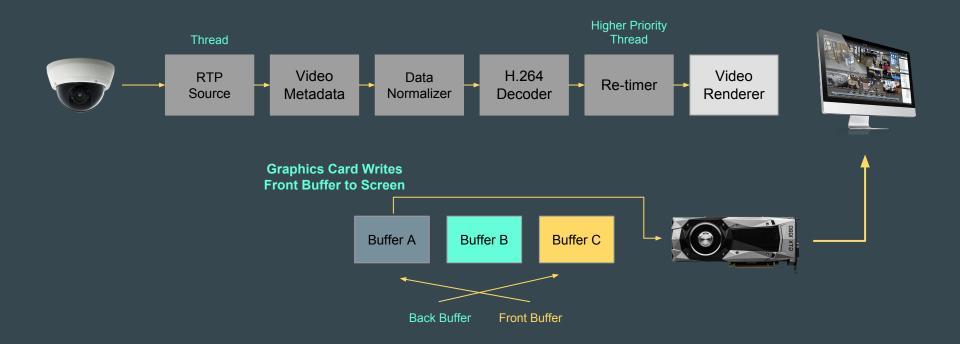


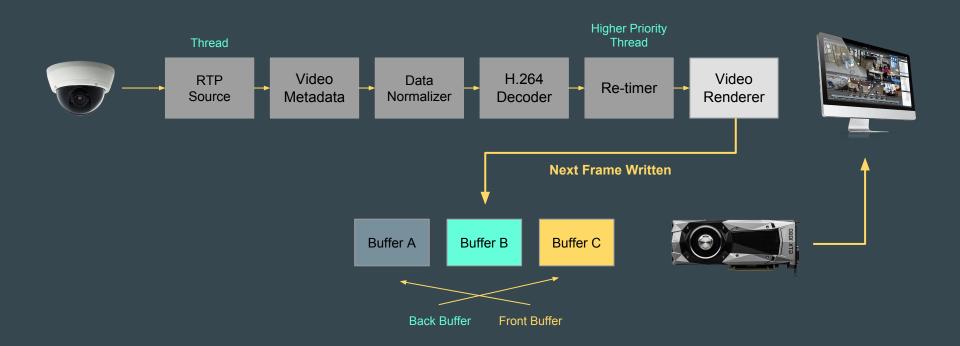


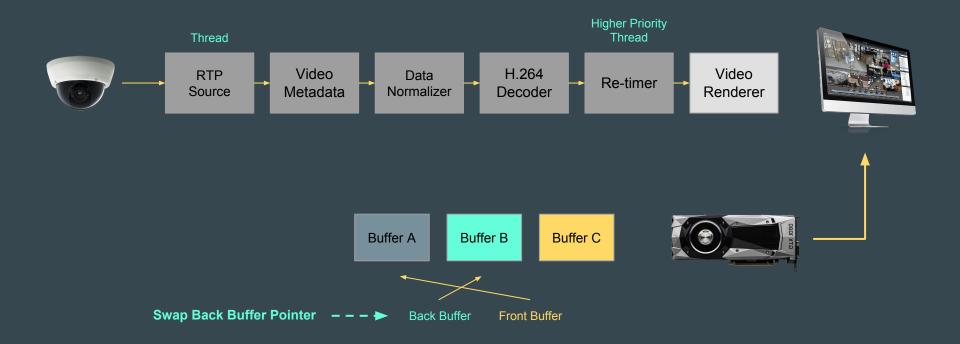


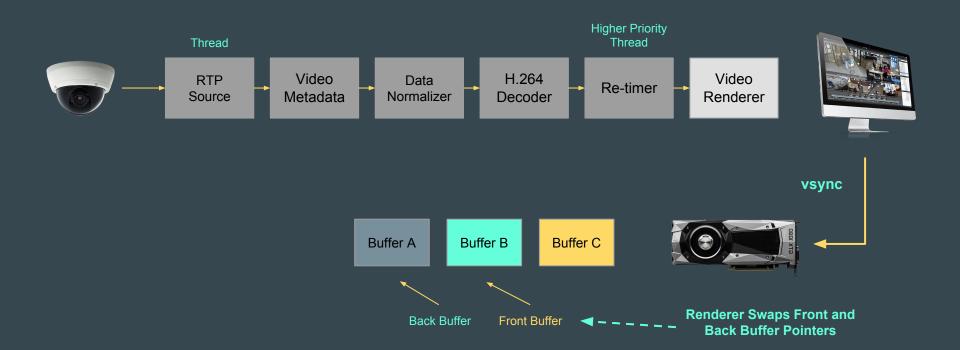


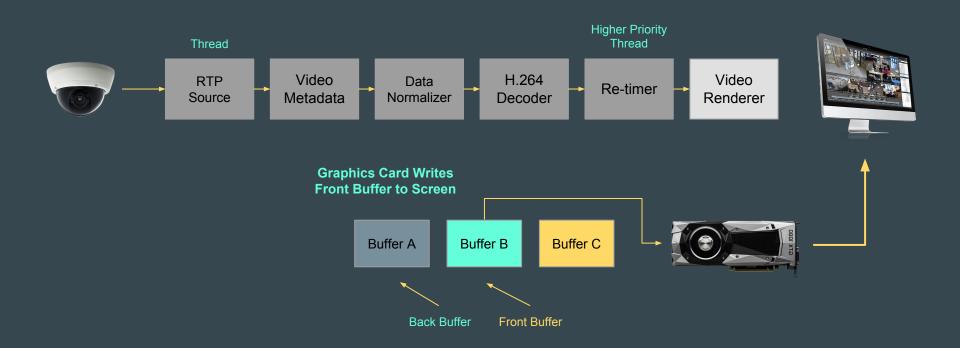




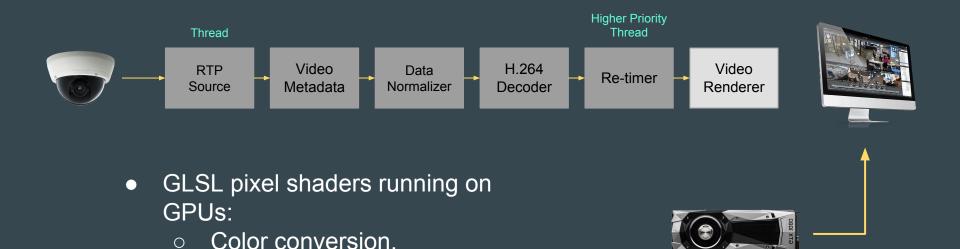








Brightness/contrast/level.



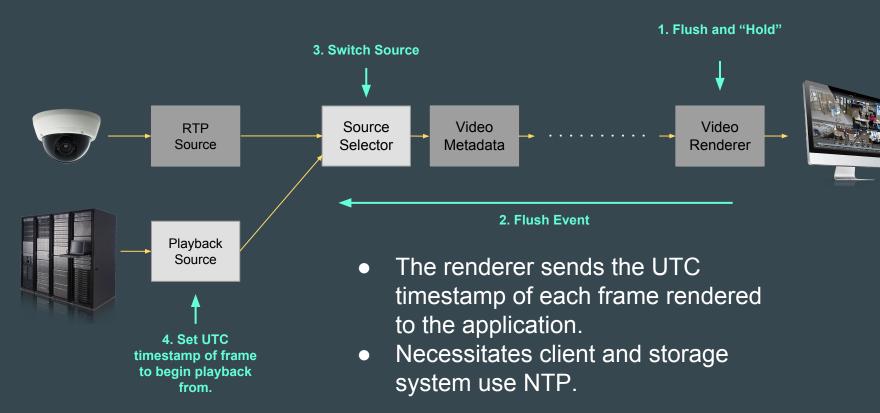


Questions so far?

happened?

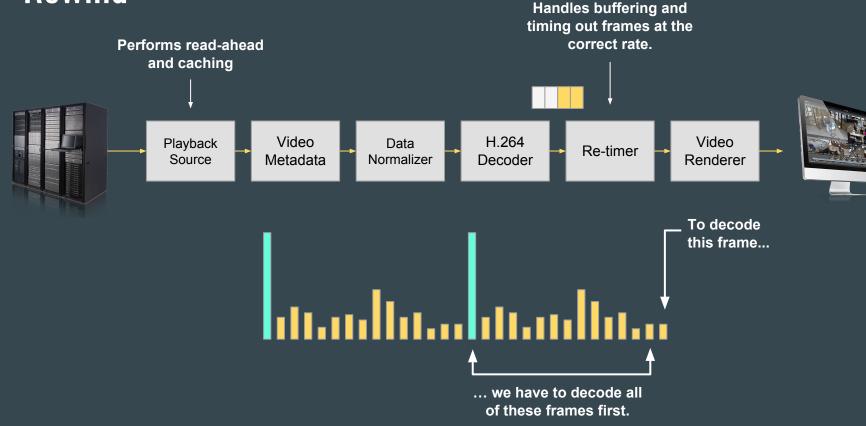
Okay... how do I review a situation that just

Live-to-Playback Transition



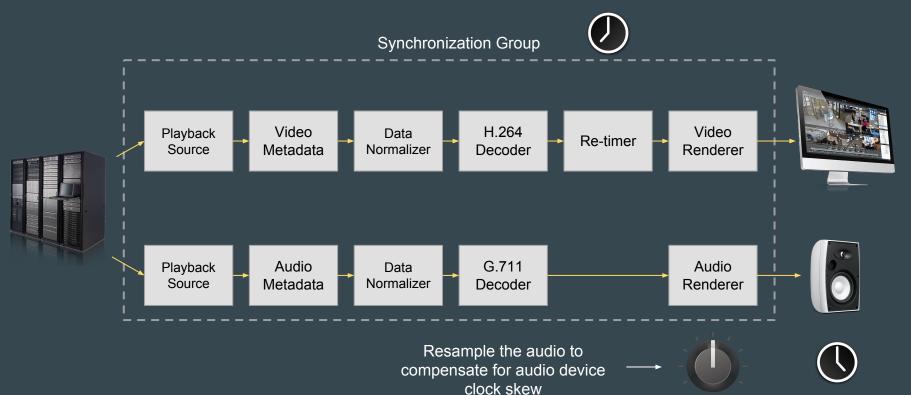
What about rewind?

Rewind



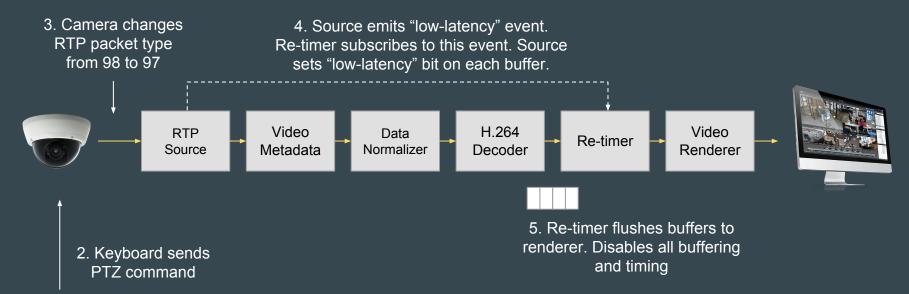
... and audio?

Audio Synchronization



Low-latency mode

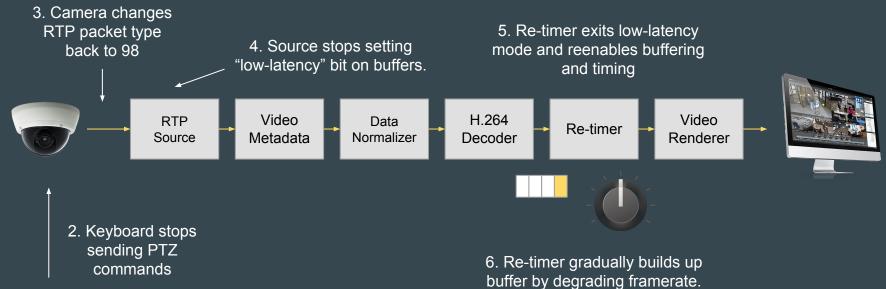
Low-latency Mode





1. User begins PTZ

Low-latency Mode



1. User ends PTZ

The End

Questions?