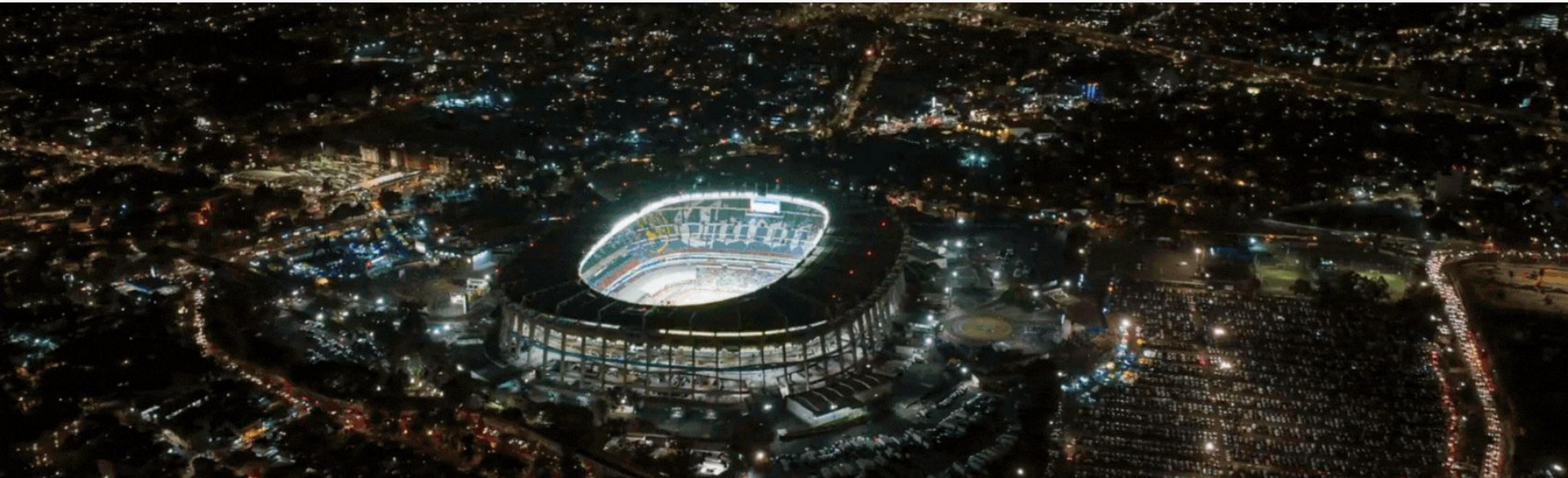




Ingesting **thousands of streams** over SRT on the same endpoint

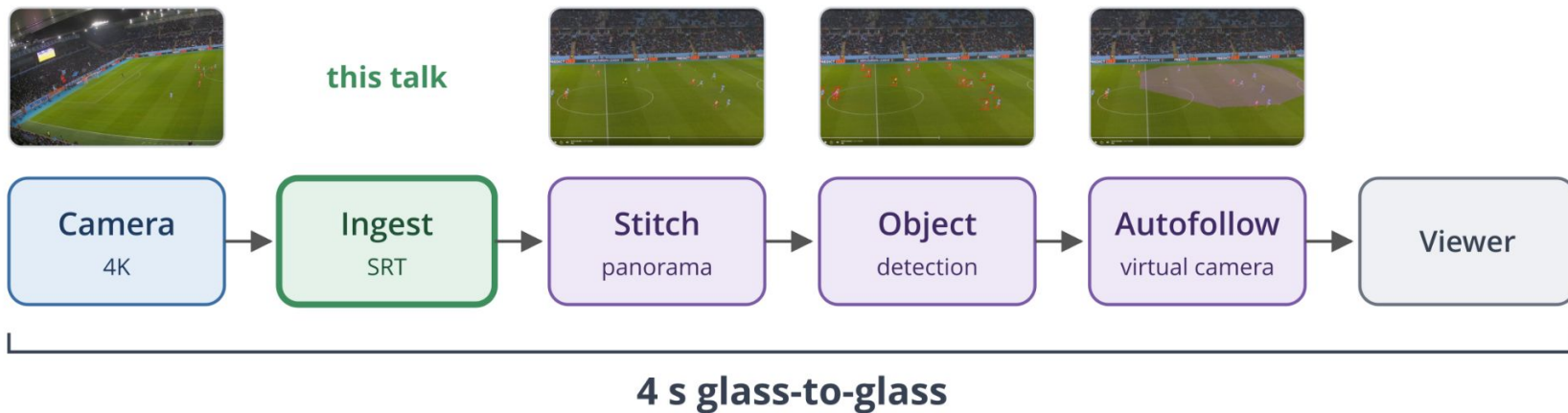
Jonas Danielsson
GStreamer Conference 2026





GSTREAMER CONFERENCE 2026

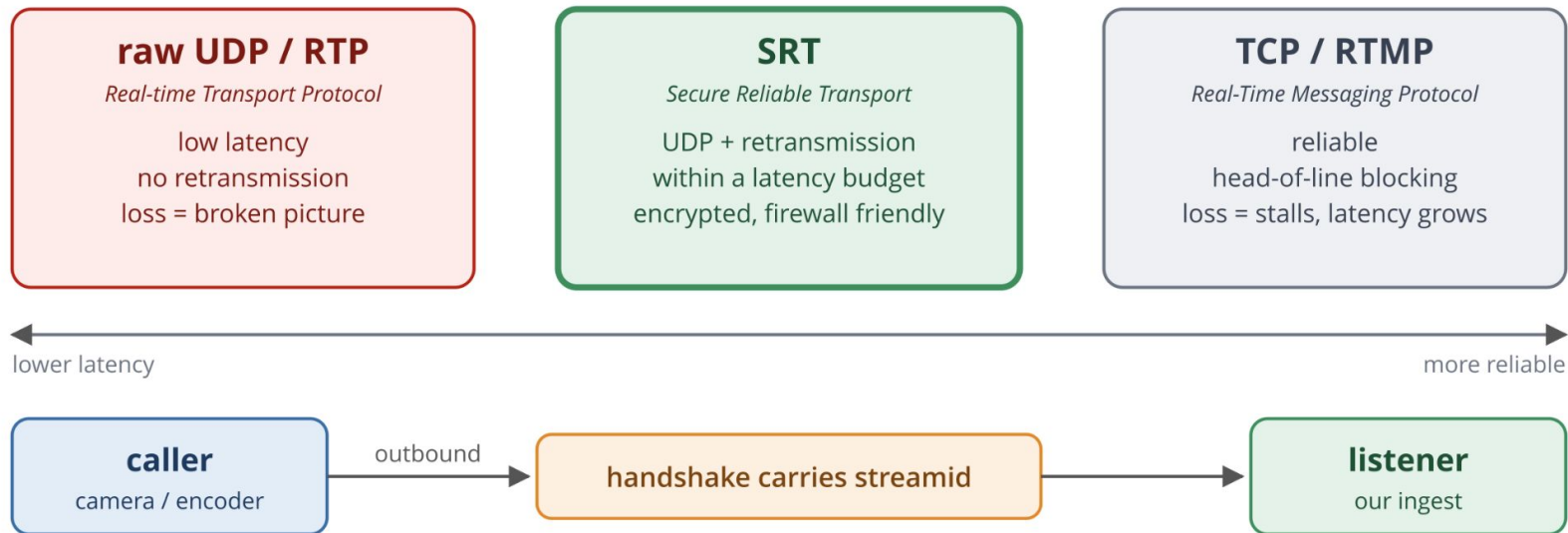
What we do: “**automate productions**”





GSTREAMER CONFERENCE 2026

SRT?

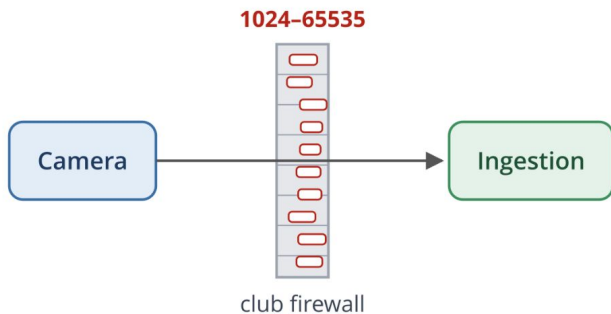




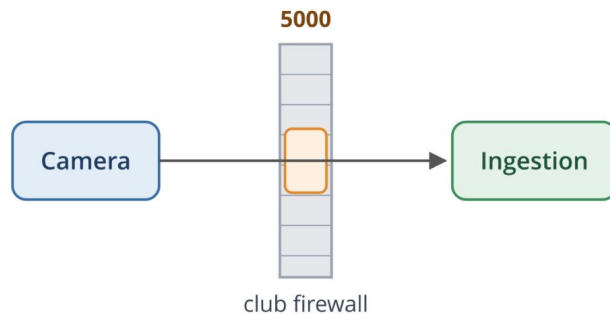
GSTREAMER CONFERENCE 2026

Why **one** port?

Before



After



Encoder settings

SRT destination

`srt://ingest:?????`

Encoder settings

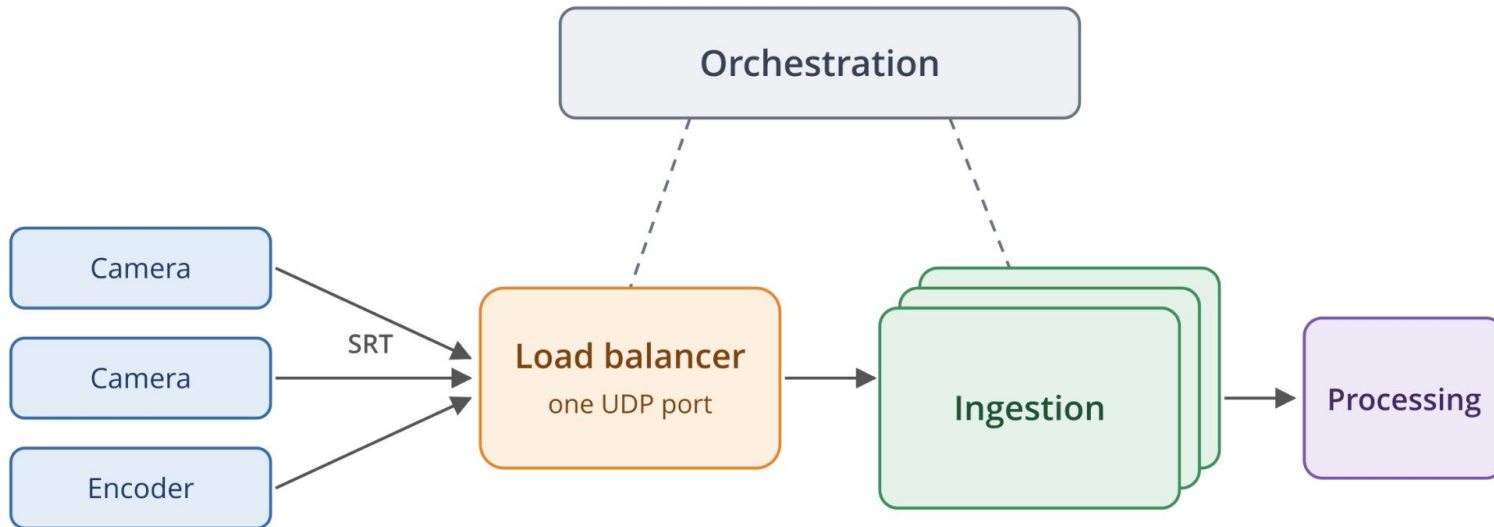
SRT destination

`srt://ingest:5000?streamid=pitch-3`



GSTREAMER CONFERENCE 2026

It takes a **village**



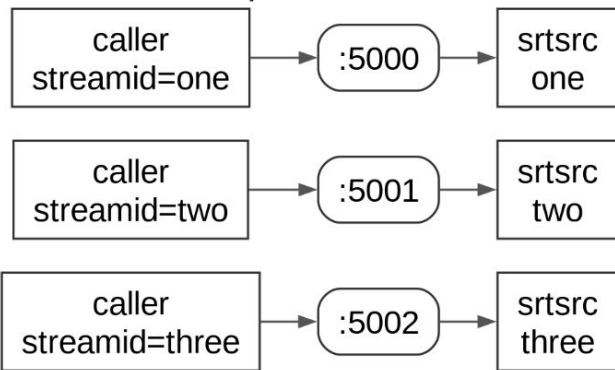


GSTREAMER CONFERENCE 2026

One port, **many streams**

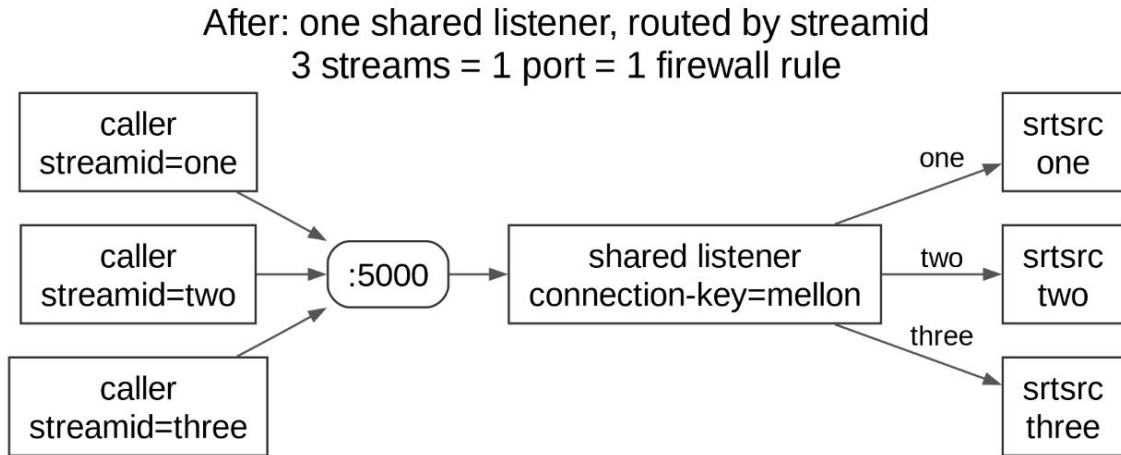
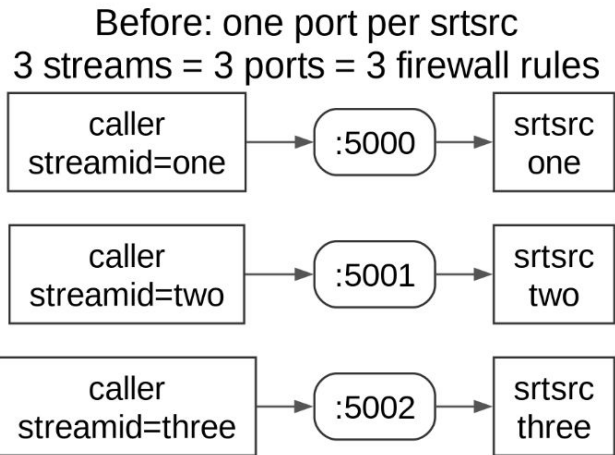
Before: one port per srtsrc

3 streams = 3 ports = 3 firewall rules





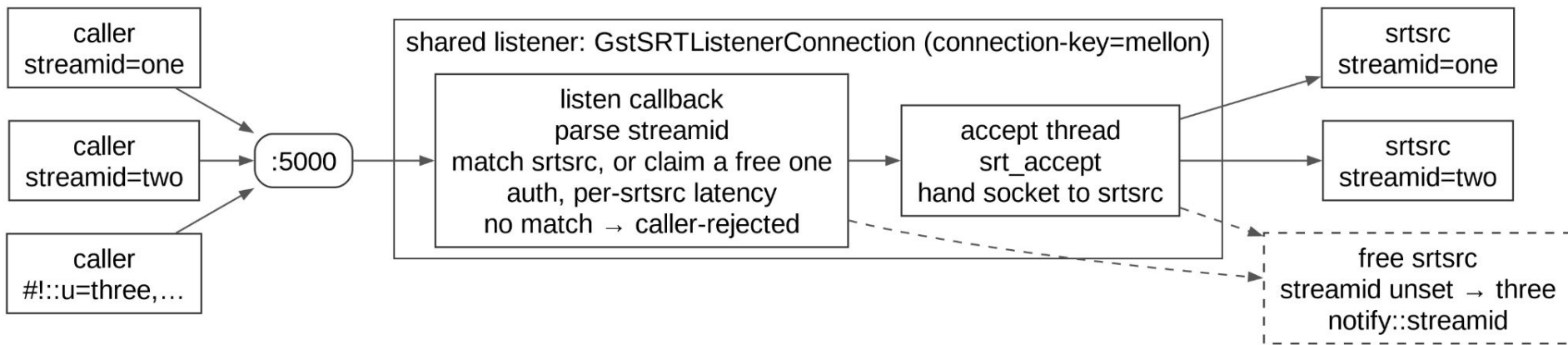
One port, **many streams**





GSTREAMER CONFERENCE 2026

One port, **many streams**





GSTREAMER CONFERENCE 2026

In **gst-launch** form

Callers

```
... ! srt sink uri="srt://host:5000?streamid=one"  
... ! srt sink uri="srt://host:5000?streamid=two"
```

Listeners

```
gst-launch-1.0 \  
  srtsrc connection-key=mellon uri="srt://:5000?mode=listener&streamid=one" \  
    ! filesink location=one.ts \  
  srtsrc connection-key=mellon uri="srt://:5000?mode=listener&streamid=two" \  
    ! filesink location=two.ts
```



GSTREAMER CONFERENCE 2026

In **gst-launch** form

Callers

```
... ! srtssink uri="srt://host:5000?streamid=three"
```

Listeners

```
gst-launch-1.0 -v \  
  srtsrc connection-key=mellon uri="srt://:5000?mode=listener" \  
  ! filesink location=out.ts
```

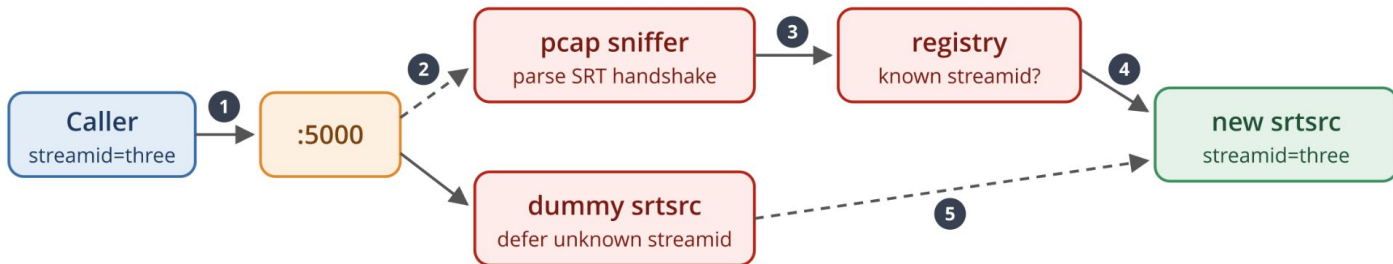
```
/GstPipeline:pipeline0/GstSRTSrc:srtsrc0: streamid = three
```



GSTREAMER CONFERENCE 2026

When do we **start** a receiver?

Our way

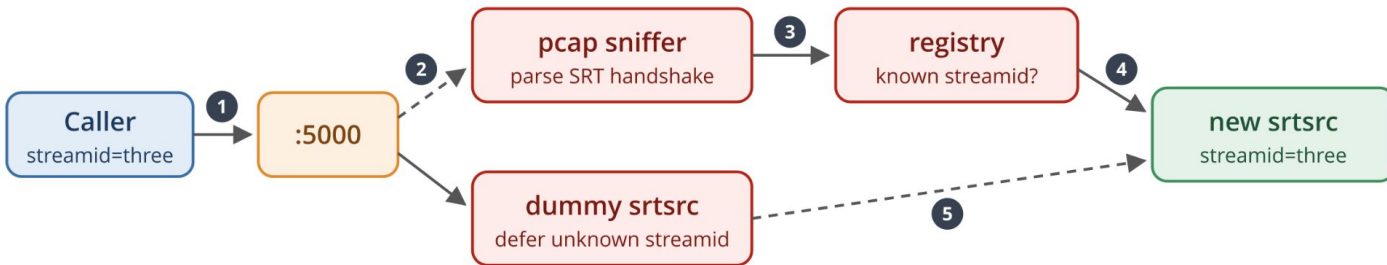




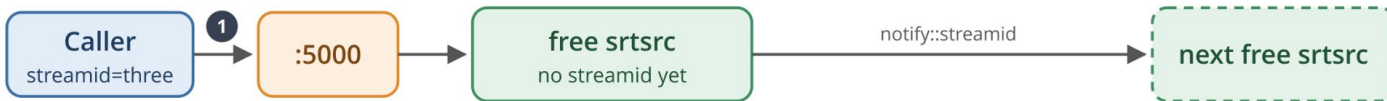
GSTREAMER CONFERENCE 2026

When do we **start** a receiver?

Our way



Mathieu's way





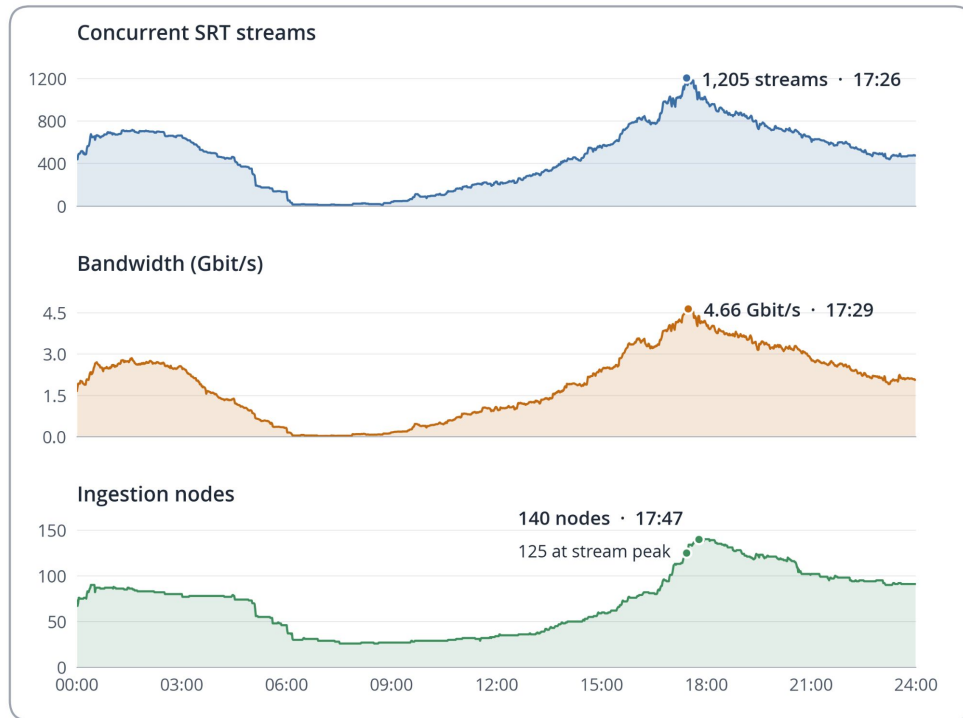
We wrote 300 lines of **Rust** to parse SRT handshakes out of pcap captures...

... Mathieu wrote a **g_object_set** and a recursive mutex.



GSTREAMER CONFERENCE 2026

- **1205 SRT streams**
- **4.66 Gbit/s**
- **140 GStreamer nodes**
- **1 UDP port**



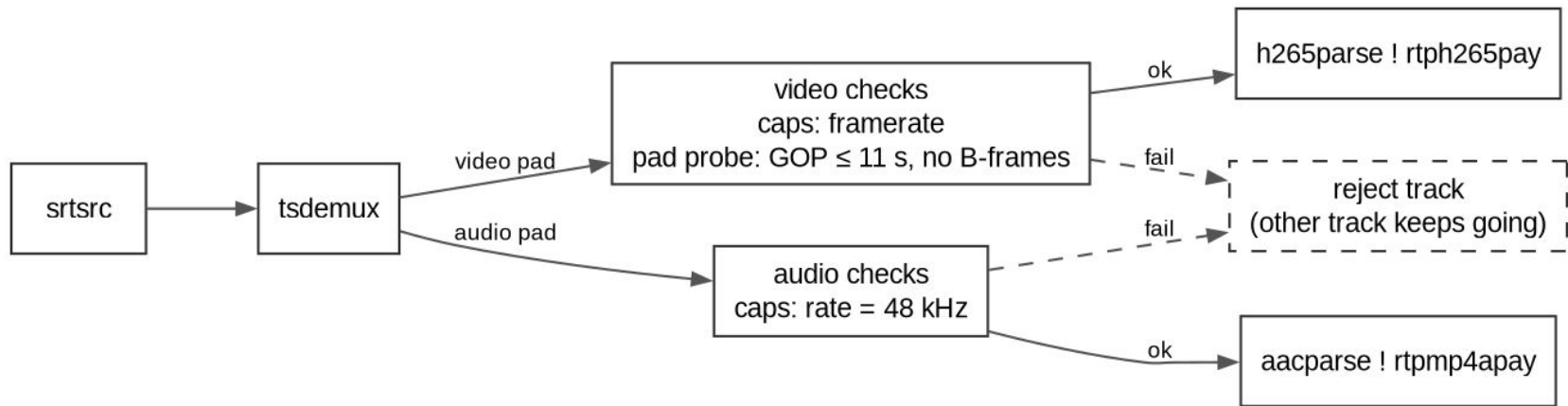


What happens after we
ingest?



GSTREAMER CONFERENCE 2026

Ingest, but **verify**





GSTREAMER CONFERENCE 2026

Tales from the fork

Upstream tsdemux

```
/* TODO: check channels, rate, and profile  
and then update caps too? */
```

```
audio/mpeg, mpegversion=4, stream-format=adts
```

With our patch

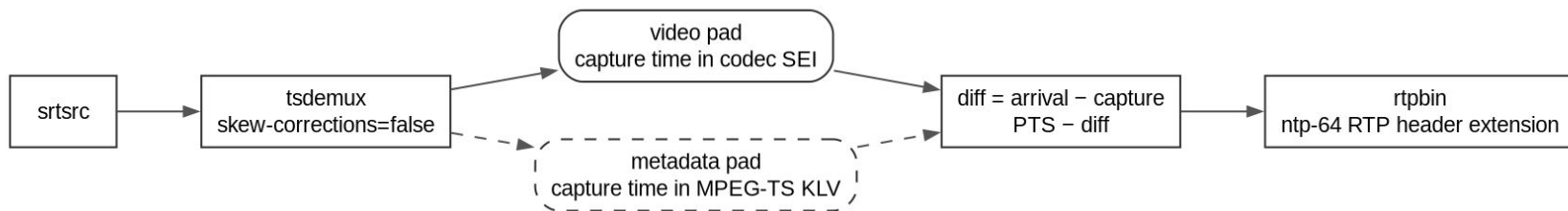
```
audio/mpeg, mpegversion=4, stream-format=adts,  
rate=44100, channels=2, profile=lc
```

[gstreamer!8507 — tsdemux: parse and expose more from ADTS header](#)



GSTREAMER CONFERENCE 2026

Fighting the PCR

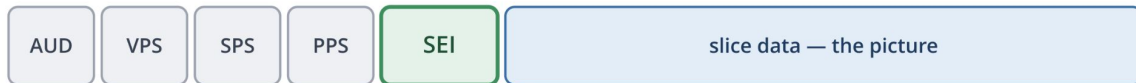




GSTREAMER CONFERENCE 2026

Supplemental Enhancement Information

One H.265 frame (access unit)



SEI user data unregistered

```
uuid 5911de00 5911de00 deadbeef baadf00d
data capture time, ns since 1900, 8 bytes
```

Ingestion

```
capture running time = sei_ntp - base_time
diff = pts - capture running time
```



Questions?