The road to Enhanced FLV and RTMP in GStreamer

Taruntej Kanakamalla

Centricular

About me

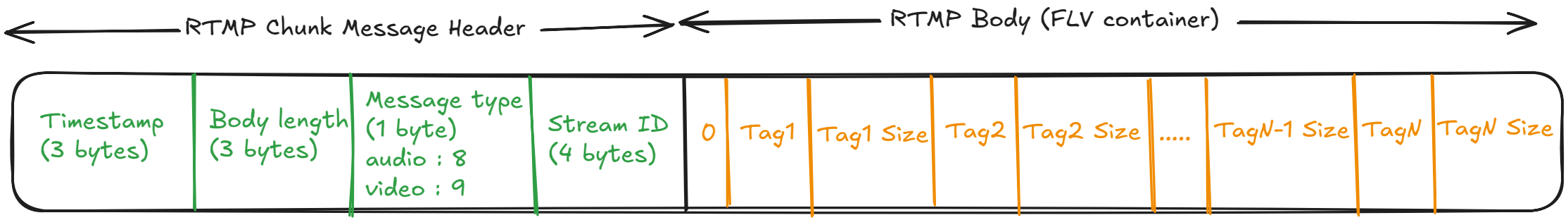
- Open-source multimedia software developer
- Currently working at Centricular
- Mostly GStreamer of late
- Little bit of PipeWire in between

Agenda

- Quick revision of RTMP and FLV protocols
- Overview of Enhanced RTMP specification
- Multitrack audio implementation
  - foreseen challenges
  - design options
  - interoperability hiccups
- Scope for other feature inclusions

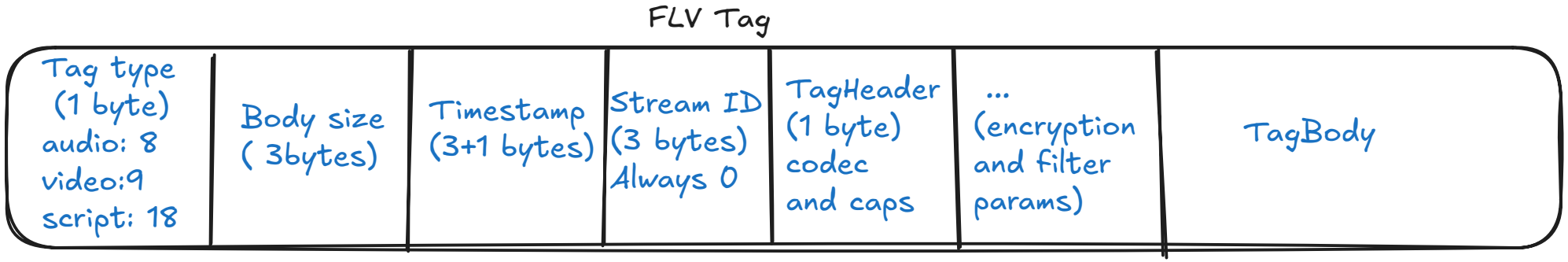
RTMP

- Real-Time Messaging Protocol
  - multiplexing and packetizing of multimedia
  - transmission over TCP
- Data sent as Chunks



FLV

- Contains a header, followed by alternating tags and size of the tags
- Each tag contains
  - metadata for audio, video, or actionscripts
  - optional encryption metadata
  - and the payload



Enhanced RTMP (eRTMP) overview

- Introduced (V2) earlier this year
- Updates the legacy protocol(s) to meet more tech standards
  - extended FLV Audio/Video TagHeader
- Reconnect request
  - connection stability and resilience

Extended FLV TagHeader

- Multitrack capabilities
  - concurrent, multiple audio and video streams
- Support for contemporary codecs
  - VP9, HEVC, Opus, FLAC etc.,
- FOURCC signalling
  - for the contemporary as well as few legacy codecs like MP3, AAC, AVC
- Multichannel audio config
- Broader range of video metadata types
- Timestamp precision - nanosecond offsets
- ModEx signal - to carry modifiers or extensions in the packets

What's existing in GStreamer?

FLV and RTMP/RTMP2 plugins

- Following the older specification
- Can handle at most one video and one audio track
- Limited (legacy) codec support

eRTMP/eFLV implementations in GStreamer

- gstreamer/gstreamer!9682 - multitrack audio in FLV muxer and demuxer (Merged)
- gstreamer/gstreamer!8398 - multitrack video in flvmux (Needs rebase)
- gstreamer/gstreamer!9806 - multitrack capability in RTMP (Draft)
- gstreamer/gstreamer!5087 - AV1 codec support to FLV (Draft)
- gstreamer/gstreamer!9873 - HEVC codec support to FLV demuxer (Brand new)

Multitrack audio

Multitrack audio

Missing pieces in the FLV plugin

- Handle multiple tracks/stream in the FLV muxer/demuxer
- Produce/consume the FLV data with extended `TagHeader`
  - minimum need track ID and FOURCC codec signaling
- Backwards compatibility
  - should still be able to stream the legacy FLV format
- Scalability for new codecs, multichannel config and other features

Multitrack audio

FLV Muxer implementation options

- Extended `flvmux`
  - Some refactoring but not complicated
  - WIP/Draft MRs already for other eRTMP features
  - Overhead of decade(s) old code
- Fresh plugin in Rust
  - Chance for better design
  - Lot of FLV infra rewrite

Multitrack audio

Extended FLV muxer

- Derive a subclass, register a new factory `eflvmux`
  - define new codecs with `audio_%u` name template
  - use the sink pad index as track ID for enhanced FLV
  - continue the inherited `audio` sink pad for legacy FLV
  - caveat: can't auto detect desired output format (enhaced or legacy)
    - specify the name template (`audio` or `audio_%u`) while requesting the sink pad
- `flvmux` continues to do only legacy FLV with single `audio` sink pad
  - thus, not breaking any existing applications

Multitrack audio

Minor interop hiccup

- Track ID semantics unclear
- Twitch considers only tracks with ID > 0
- Debugged using packet dissections and OBS code

Multitrack audio

Extended FLV demuxer

- `src` pad templates similar to muxer's `sink` pads
  - `audio` for legacy and `audio_%u` for enhanced
- No pad confusion unlike the muxer
  - because, all src pads output demuxed audio
- Pads are named after the track IDs

Multitrack audio

Extended FLV demuxer

- `GstStreamCollection` contains streams for all the tracks
- Track ID <=> src pad index <=> stream ID
- Enhanced FLV track with lowest ID will be the default stream/track

Enhanced RTMP

Future plans

- All structural changes likely done
- Works in progress
  - gstreamer/gstreamer!9806 - multitrack capability in RTMP sink/src
  - gstreamer/gstreamer!8398 - multitrack video in flvmux
  - gstreamer/gstreamer!5087 - AV1 codec support to FLV
  - gstreamer/gstreamer!9873 - HEVC codec support to FLV demuxer
- Add other codecs support
- Include more features as needed

Thank You!