Writing an MP4 Muxer for Fun and Profit

Except there is no Profit, only Pain.

Dennis Sädtler - FOSDEM 2025

Who am I?

- Software Developer from Germany
- Currently doing Media Stuff™ at Twitch
- One of the main contributors to OBS Studio since ~2020
- Also contributed to FFmpeg and other FOSS projects
- Knows what an MP4 is

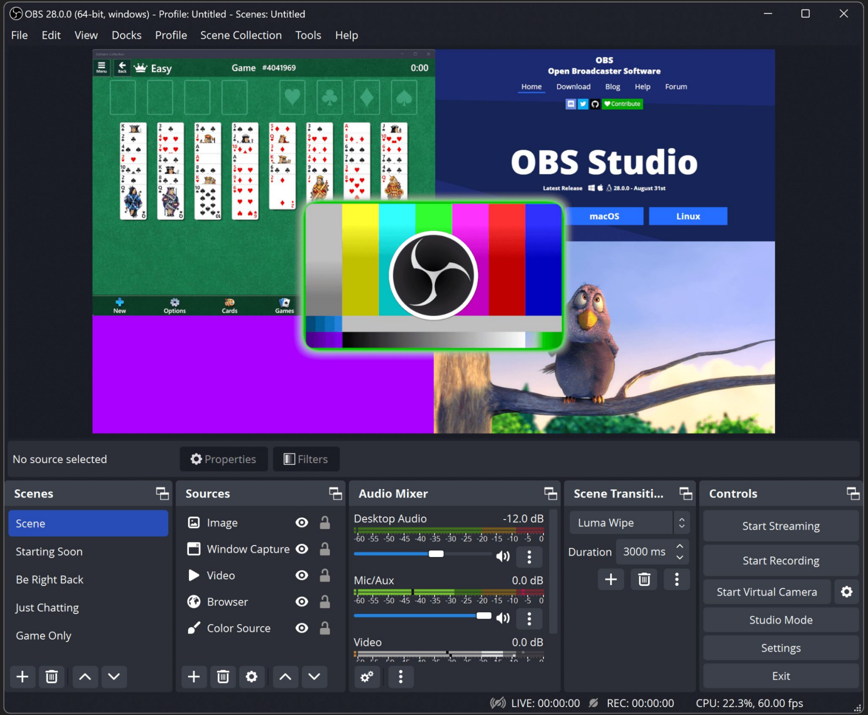
What is MP4

Probably the most well known and widely used media “container” for audio and video streams

- Apple created the “QuickTime File Format” in the 90s
- Adapted by the ISO into MP4 in 2001
- Later generalised into ISO-BMFF (Base Media File Format) and CMAF

What is OBS Studio?

- FLOSS (GPLv2+) software for live streaming and recording
- First-party packages for Windows, Linux, and macOS
- 10s of Millions Monthly Active Users
  - Massive variation in technical skill level
- Mostly used without any backup recording
  - We need to avoid data loss as much as possible!
- More than just gaming!
  - Teachers prepping online lessons
  - Doctors recording ultrasound machines
  - etc.



OBS 28.0.0 (64-bit, windows) - Profile: Untitled - Scenes: Untitled

No source selected

Desktop Audio -12.0 dB

Mic/Aux 0.0 dB

Video 0.0 dB

Luma Wipe

Duration 3000 ms

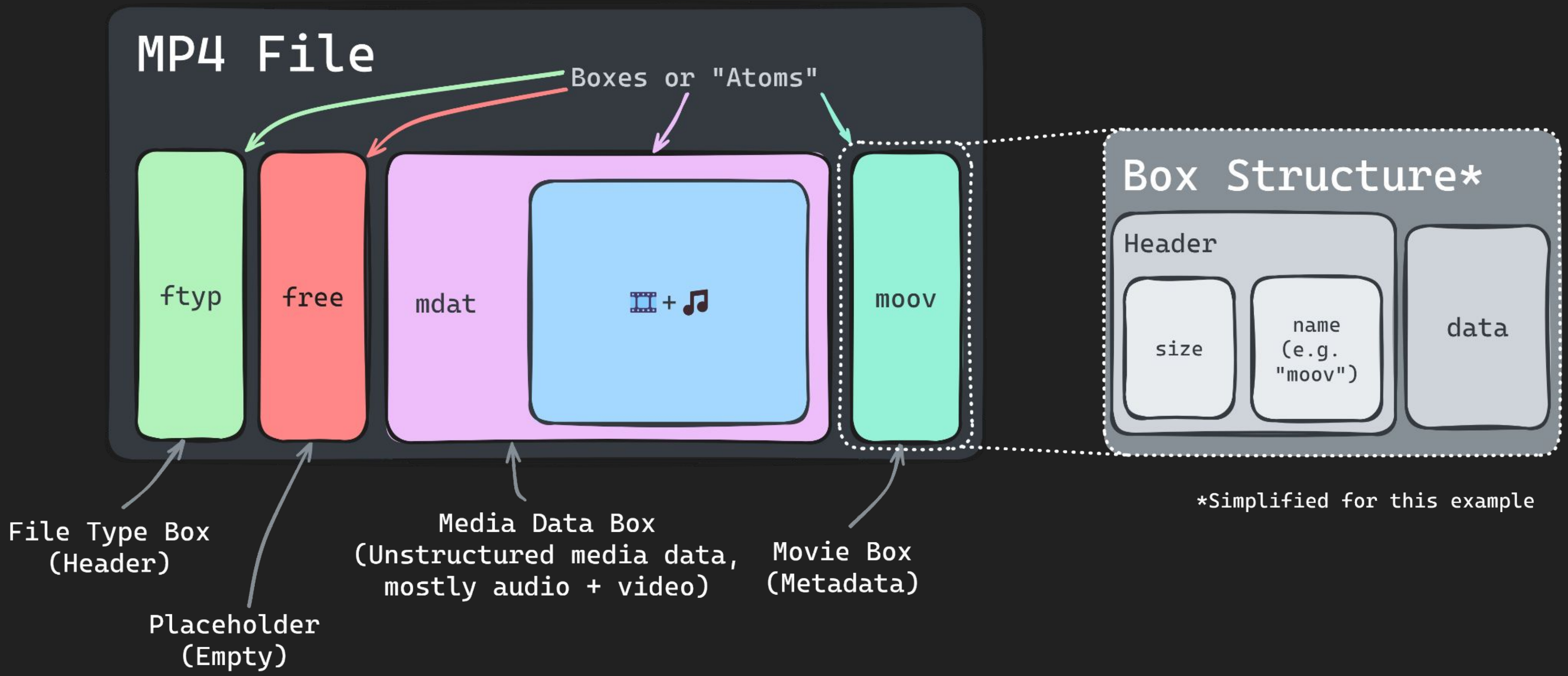
LIVE: 00:00:00 REC: 00:00:00 CPU: 22.3%, 60.00 fps

The Conundrum

To produce files useful to most users we **need** to use MP4

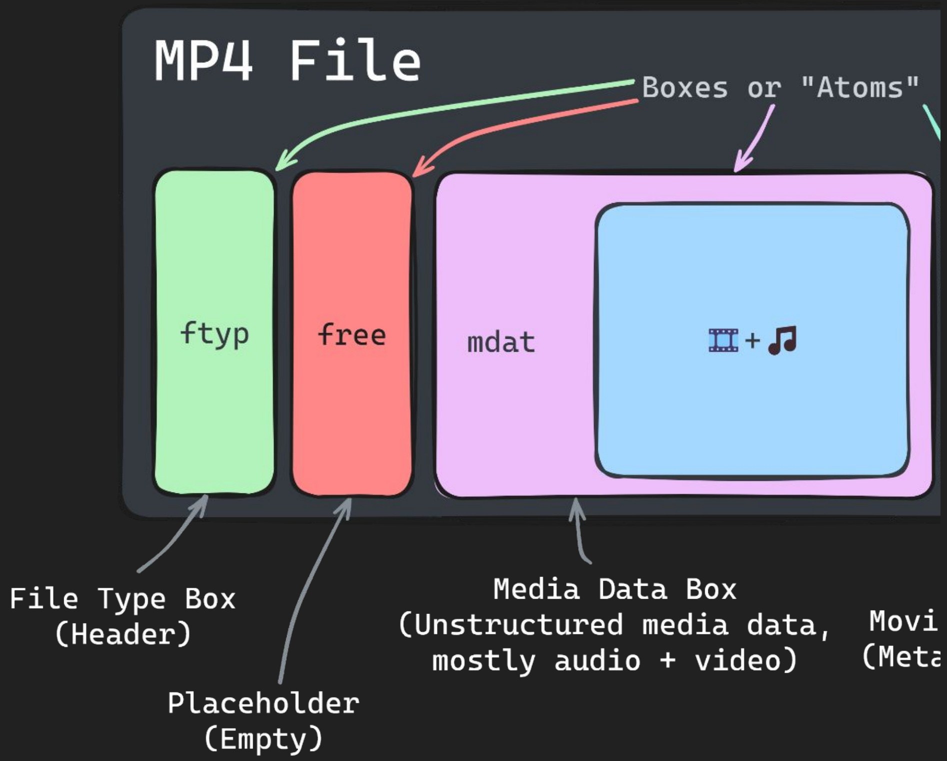
To avoid data loss we **cannot** use MP4

Why plain MP4 is a Problem



*Simplified for this example

Why plain MP4 is a Problem



MP4 File

Boxes or "Atoms"

ftyp

free

mdat

🎞 + ♫

File Type Box
(Header)

Placeholder
(Empty)

Media Data Box
(Unstructured media data,
mostly audio + video)

Movi
(Meta

So what have we been doing?

Until 2019

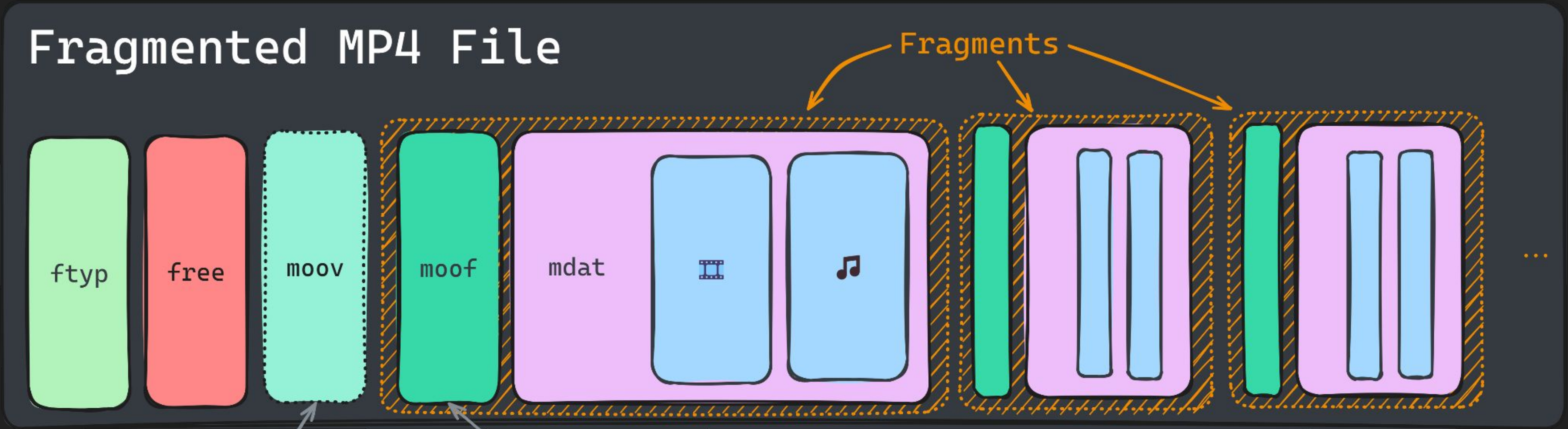
(I can't believe it either)

Since 2019

The Problem

omputer
usines
Center
Do I look like I know what a
MKV file is?

Fragmented MP4 to the Rescue?



Incomplete Movie Box
(No information about A/V data)

Movie Fragment Box
(Fragment Metadata)

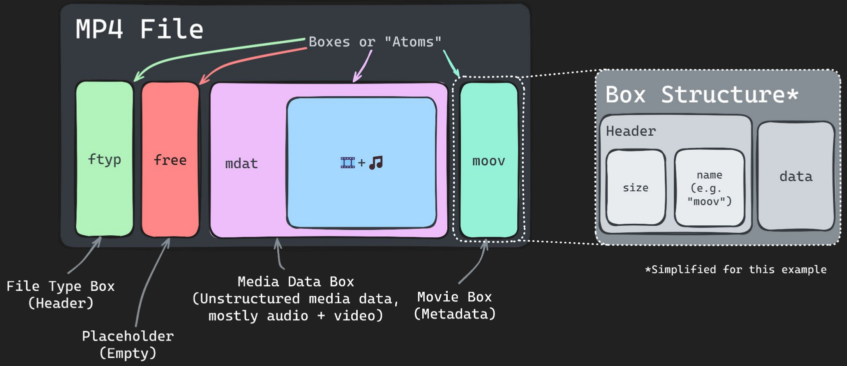
nope.avi

It’s not *just* Windows…

- Not supported by some software
  - e.g. older versions of Resolve crash to desktop
- Can be slow to open from HDD/network shares
  - Each moof needs to be read to obtain duration
- Embeds don’t work in Discord etc.

Result: Just not a good user experience

What if...?



MP4 File

Boxes or "Atoms"

ftyp

free

mdat

moov

Box Structure*

Header

size

name (e.g. "moov")

data

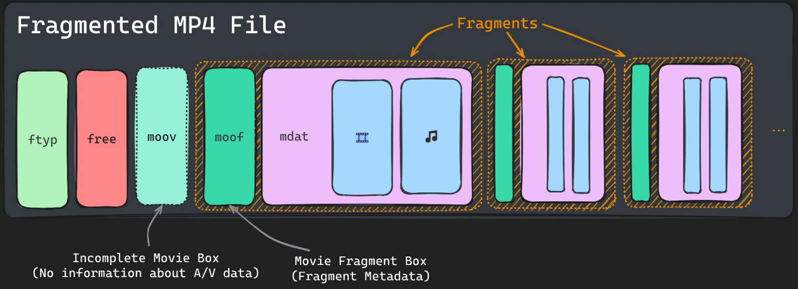
File Type Box (Header)

Placeholder (Empty)

Media Data Box (Unstructured media data, mostly audio + video)

Movie Box (Metadata)

*Simplified for this example



Fragmented MP4 File

Fragments

ftyp

free

moov

moof

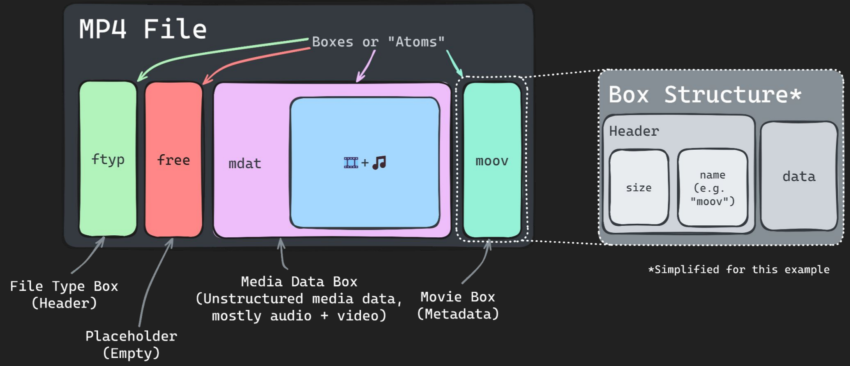
mdat

...

Incomplete Movie Box (No information about A/V data)

Movie Fragment Box (Fragment Metadata)

What if...?



MP4 File

Boxes or "Atoms"

ftyp

free

mdat

moov

File Type Box
(Header)

Placeholder
(Empty)

Media Data Box
(Unstructured media data,
mostly audio + video)

Movie Box
(Metadata)

Box Structure*

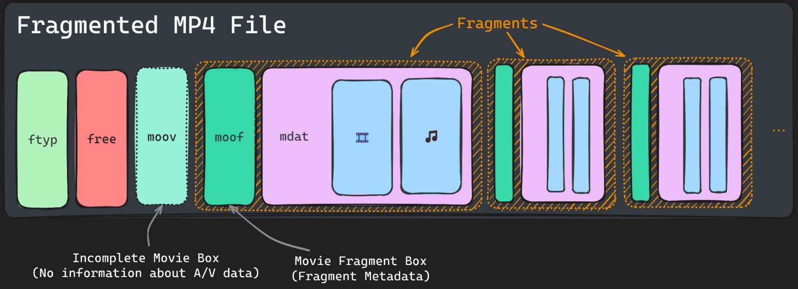
Header

size

name
(e.g.
"moov")

data

*Simplified for this example



Fragmented MP4 File

Fragments

ftyp

free

moov

moof

mdat

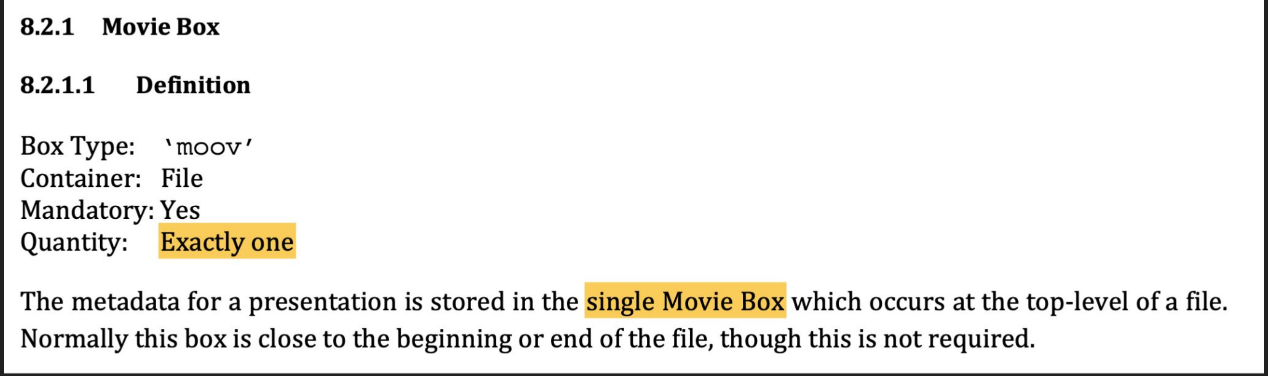
...

Incomplete Movie Box
(No information about A/V data)

Movie Fragment Box
(Fragment Metadata)

ISO/IEC:

What if…?



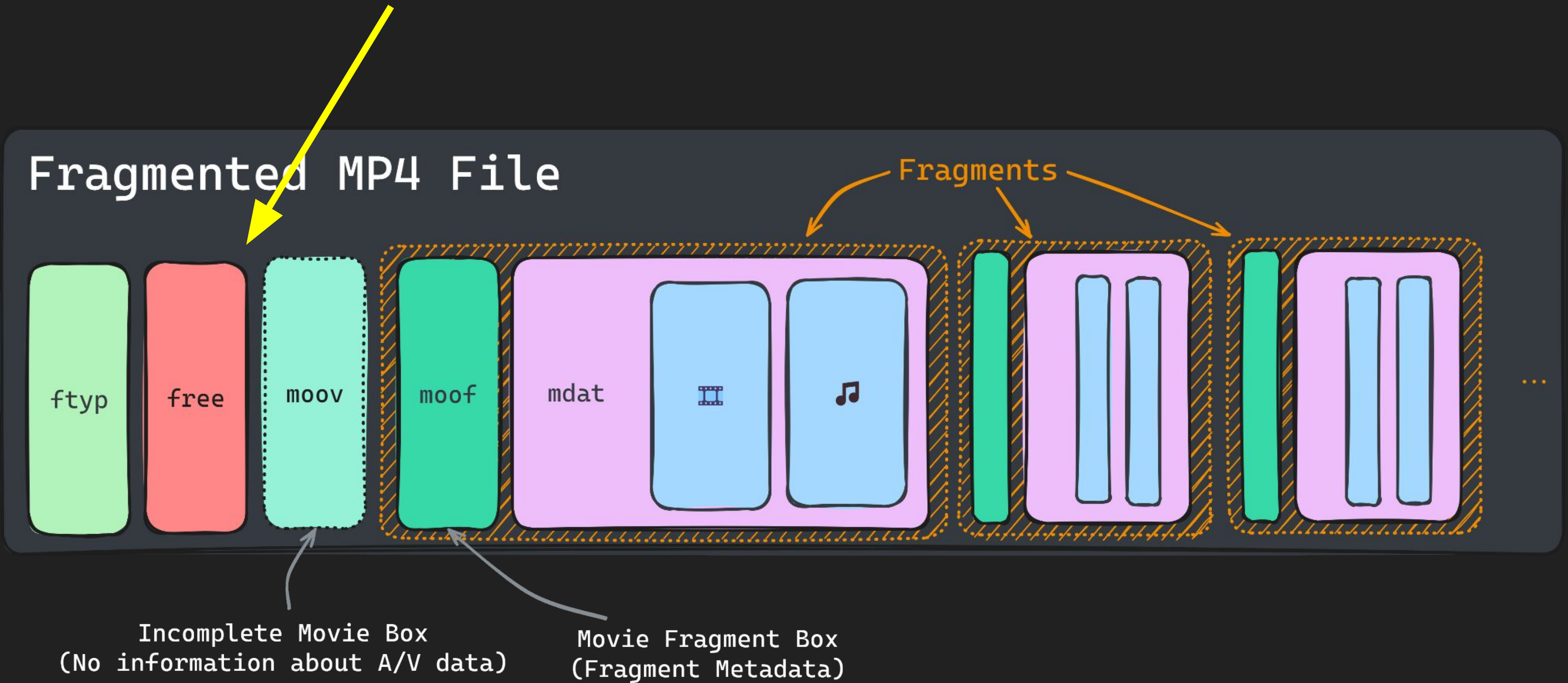
8.2.1 Movie Box

8.2.1.1 Definition

Box Type: `'moov'`
Container: File
Mandatory: Yes
Quantity: Exactly one

The metadata for a presentation is stored in the single Movie Box which occurs at the top-level of a file. Normally this box is close to the beginning or end of the file, though this is not required.

ISO/IEC:

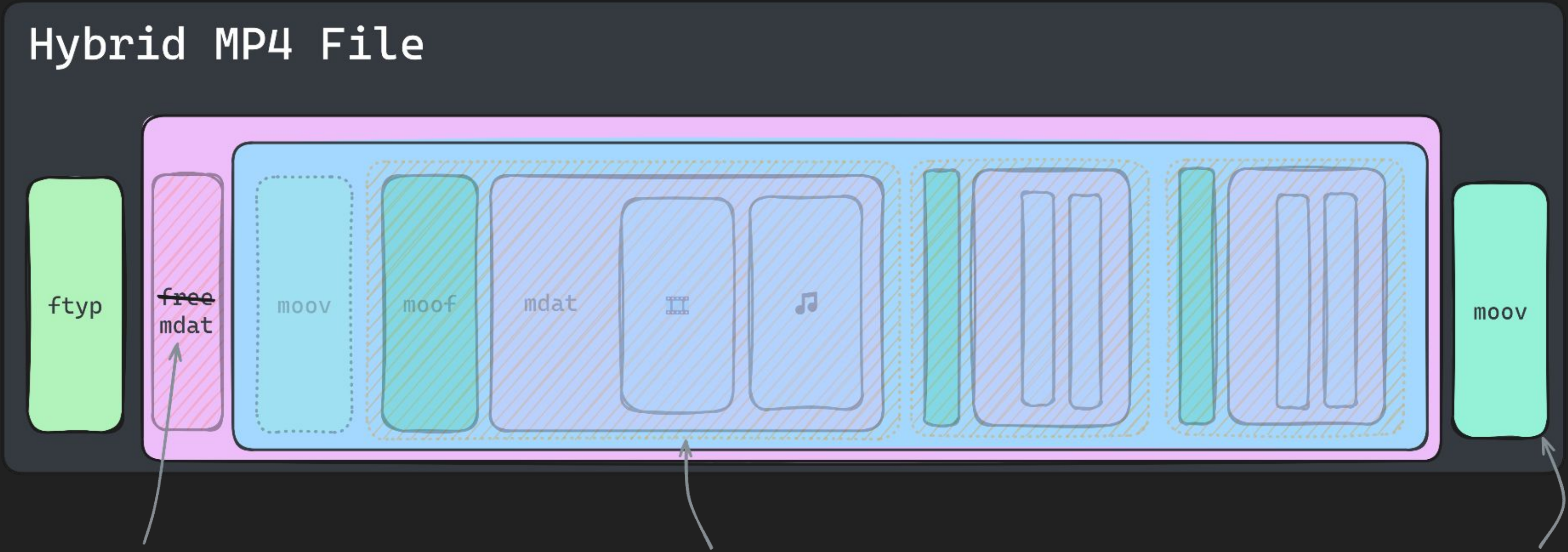


Fragmented MP4 File

Incomplete Movie Box
(No information about A/V data)

Movie Fragment Box
(Fragment Metadata)

Enter “Hybrid MP4”



Placeholder box header has been overwritten to become a large Media Data Box

Fragmented MP4 structure is hidden inside mdat box

Regular Complete Movie Box

Why write a new muxer?

Upstreaming to FFmpeg had some risks

- Might not be accepted due to “niche” use case
- Could take months to be merged, even longer before it is in a stable release
- We ship Ubuntu LTS packages and would like to have this feature before 2028

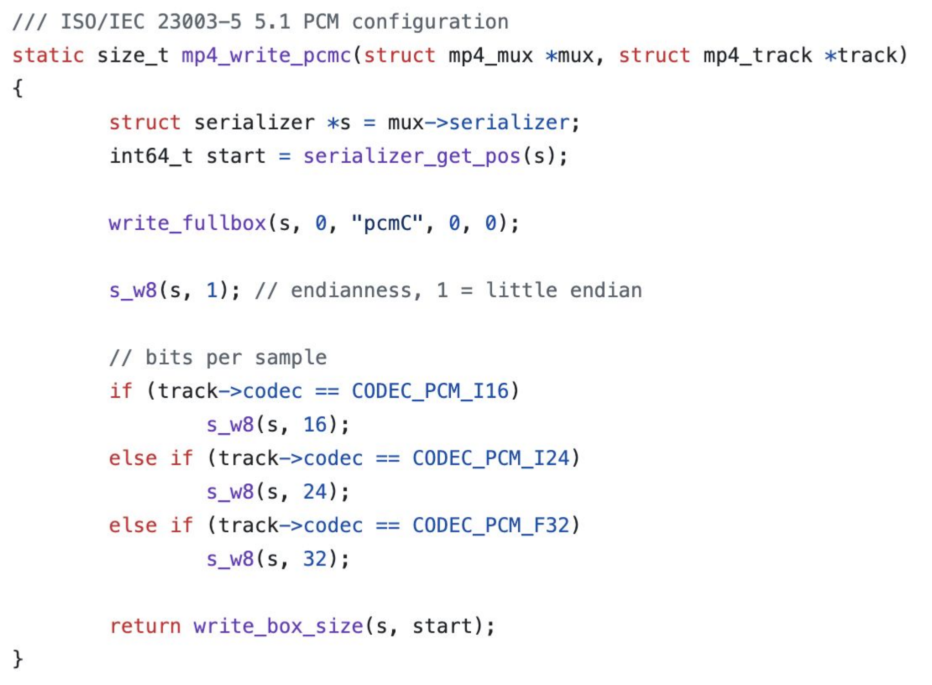
Doing it natively provides much greater control and room for experimentation

It might be fun.

And I could learn something…

How I went about it…

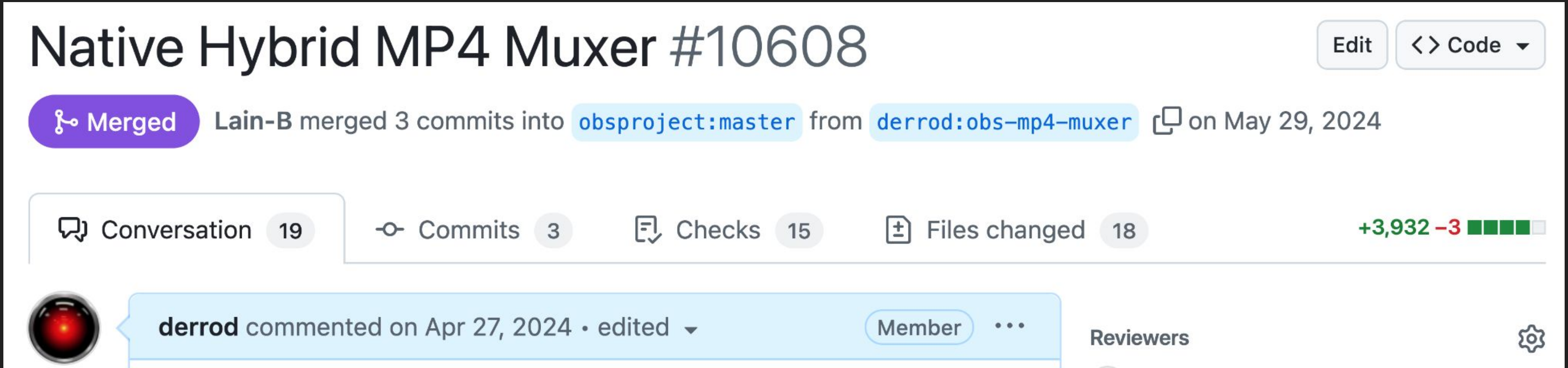
- Follows similar structure to FFmpeg MOV/MP4 muxer
  - Simplicity FTW
- Based on ISO spec, but some parts used FFmpeg or GPAC as reference
- Strictly limited to features relevant to OBS
  - Only built-in audio/video codecs
  - Can only do fragmentation on keyframes
- Some features new to OBS
  - Chapter Markers
  - Additional Metadata



```c
/// ISO/IEC 23003-5 5.1 PCM configuration
static size_t mp4_write_pcmc(struct mp4_mux *mux, struct mp4_track *track)
{
        struct serializer *s = mux->serializer;
        int64_t start = serializer_get_pos(s);

        write_fullbox(s, 0, "pcmC", 0, 0);

        s_w8(s, 1); // endianness, 1 = little endian

        // bits per sample
        if (track->codec == CODEC_PCM_I16)
                s_w8(s, 16);
        else if (track->codec == CODEC_PCM_I24)
                s_w8(s, 24);
        else if (track->codec == CODEC_PCM_F32)
                s_w8(s, 32);

        return write_box_size(s, start);
}
```

End Result

- Muxer is ~2900 lines
- Initial PoC and implementation took a few days
- Testing & Review took a month



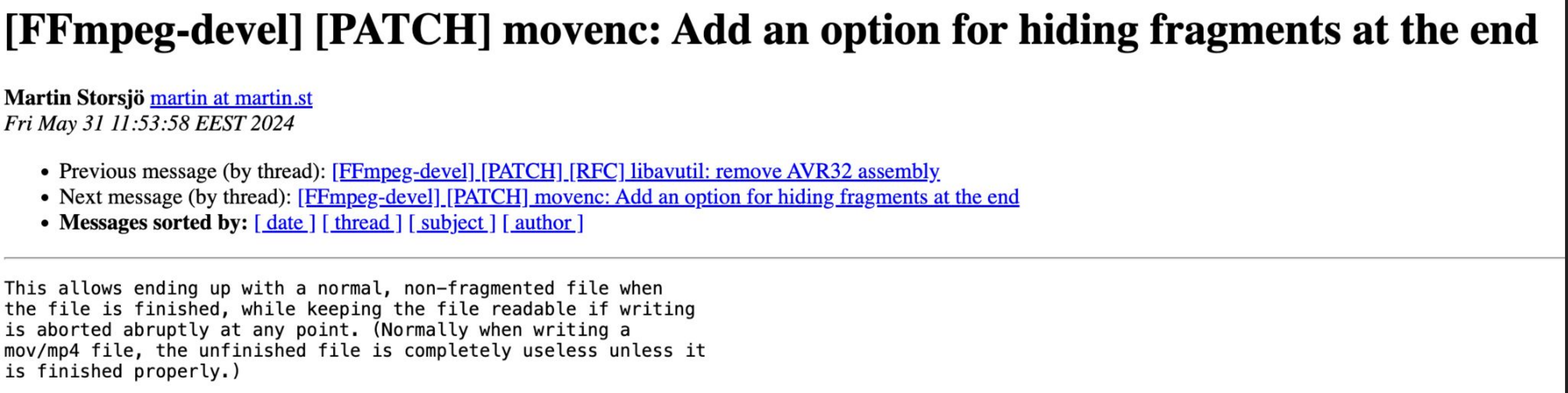
Native Hybrid MP4 Muxer #10608

Merged Lain-B merged 3 commits into obsproject:master from derrod:obs-mp4-muxer on May 29, 2024

derrod commented on Apr 27, 2024 • edited

FFmpeg can do it, too

- 2 Days after we merged it into OBS, Martin Storsjö submits a similar feature to FFmpeg
  - Pure coincidence!
- Available via `-movflags hybrid_fragmented`



[FFmpeg-devel] [PATCH] movenc: Add an option for hiding fragments at the end

Martin Storsjö martin at martin.st
Fri May 31 11:53:58 EEST 2024

- Previous message (by thread): [FFmpeg-devel] [PATCH] [RFC] libavutil: remove AVR32 assembly
- Next message (by thread): [FFmpeg-devel] [PATCH] movenc: Add an option for hiding fragments at the end
- **Messages sorted by:** [date] [thread] [subject] [author]

```
This allows ending up with a normal, non-fragmented file when
the file is finished, while keeping the file readable if writing
is aborted abruptly at any point. (Normally when writing a
mov/mp4 file, the unfinished file is completely useless unless it
is finished properly.)
```

Possibly Cursed Ideas for the Future

Actually make it “Hybrid”!

- Fragmented for Streaming
  - HLS/DASH byte range pointing at header/fragments in “hidden” section
- Progressive for Download

Server-side concatenation

- Create new header/footer, then stitch fragments together into one progressive file
- Some Object Storage providers support this, e.g. Google Cloud Storage
- Could also be done on-the-fly!

More to Come!

- Hybrid MOV is almost ready for OBS 31.1 later this year
    - ProRes and Apple-flavour PCM support
- Timecode track
- And more (but not as interesting) things!

Lessons Learned

Audio is hard

- Discovered bugs related to “priming samples” in OBS that has existed forever
  - Minor audio desync for Opus and non-default AAC encoders
  - Also found a related bug in FFmpeg (#11031)

Video is hard, too

- Frame-reordering was created by the devil

But MP4 is pretty simple!

Days of hard work can be summed up in 5 minutes and that hurts

Thanks & Acknowledgements

FFmpeg and its contributors for documenting the undocumented!

GPAC for mp4box.js, which has been invaluable for debugging my muxer and inspecting its output

Apple's old QTFF documentation for actually being quite good and having great explanations for concepts such as priming samples

NOT the ISO for paywalling these specs and making it a goddamn paperchase where every time you get one document it references three others that are also paywalled

Fin.

Feedback / Questions / Complaints about Enhanced RTMP:

https://socials.rodney.io/