Exploring the Capabilities of ISOBMFF (MP4)

ACM MILE-HIGH VIDEO 2023

Dimitri Podborski

The ISO Base Media File Format

ISO/IEC 14496-12; MPEG-4 Part 12

What has been told so far?

- A LOT! The spec is over 25 years old!

Many talks have been given. Such as:

- Mile High Video 2018: “An overview of the ISO Base Media File Format”
- Mile High Video 2022: “Introduction to ISOBMFF and CMAF”
- Demuxed 2015: “What is in the box?”

Many resources can be found on the Internet

Agenda

History (with additional insights)

ISOBMFF impact, tools, conformance, 4CC registration

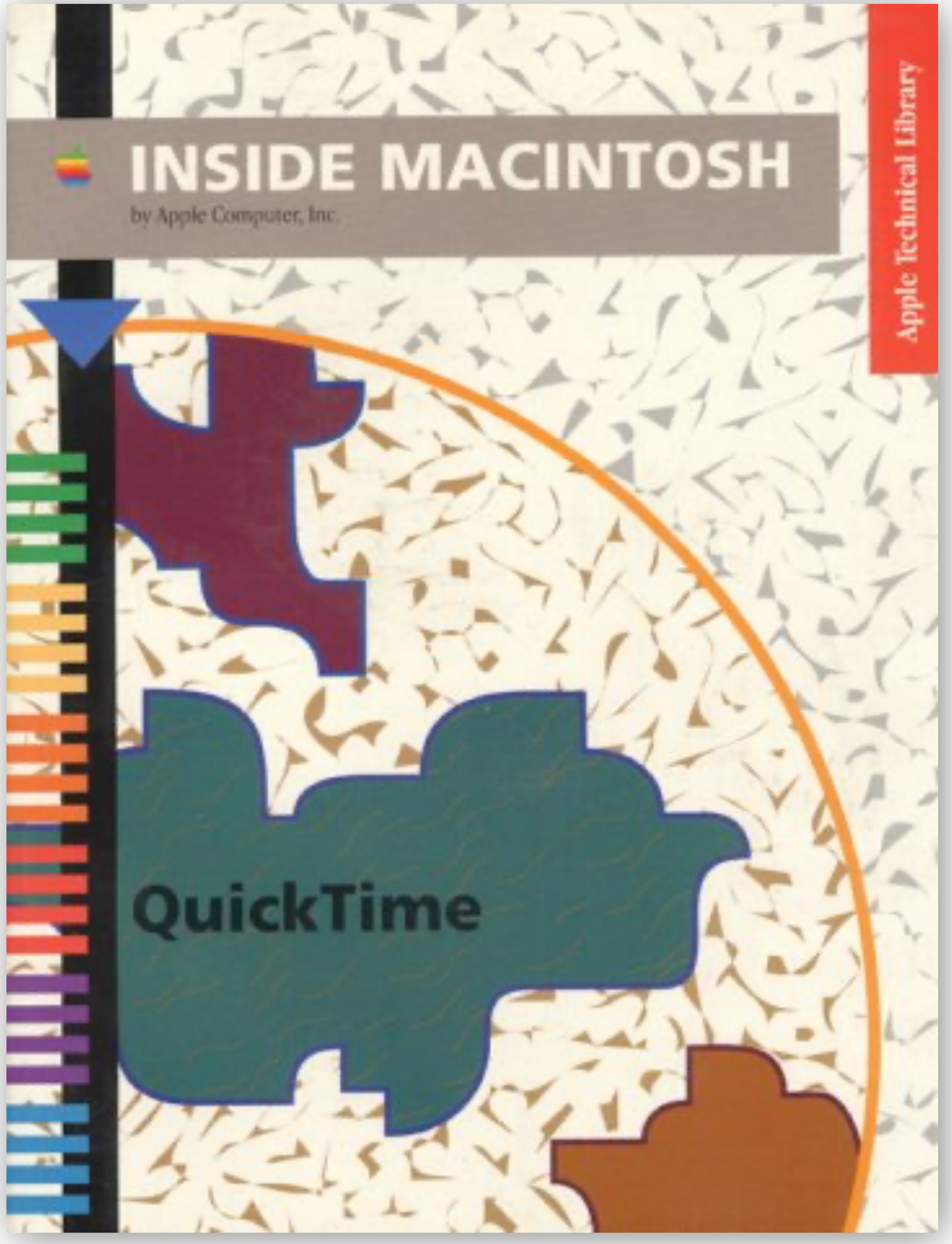
ISOBMFF basics and existing tools defines in the spec

Go into more details on some topics not covered so far:

- Sample Groups (Compact / Fragmented / Defragmentation)
- Multiplexed timed metadata tracks

Inside Macintosh

QuickTime (1993)



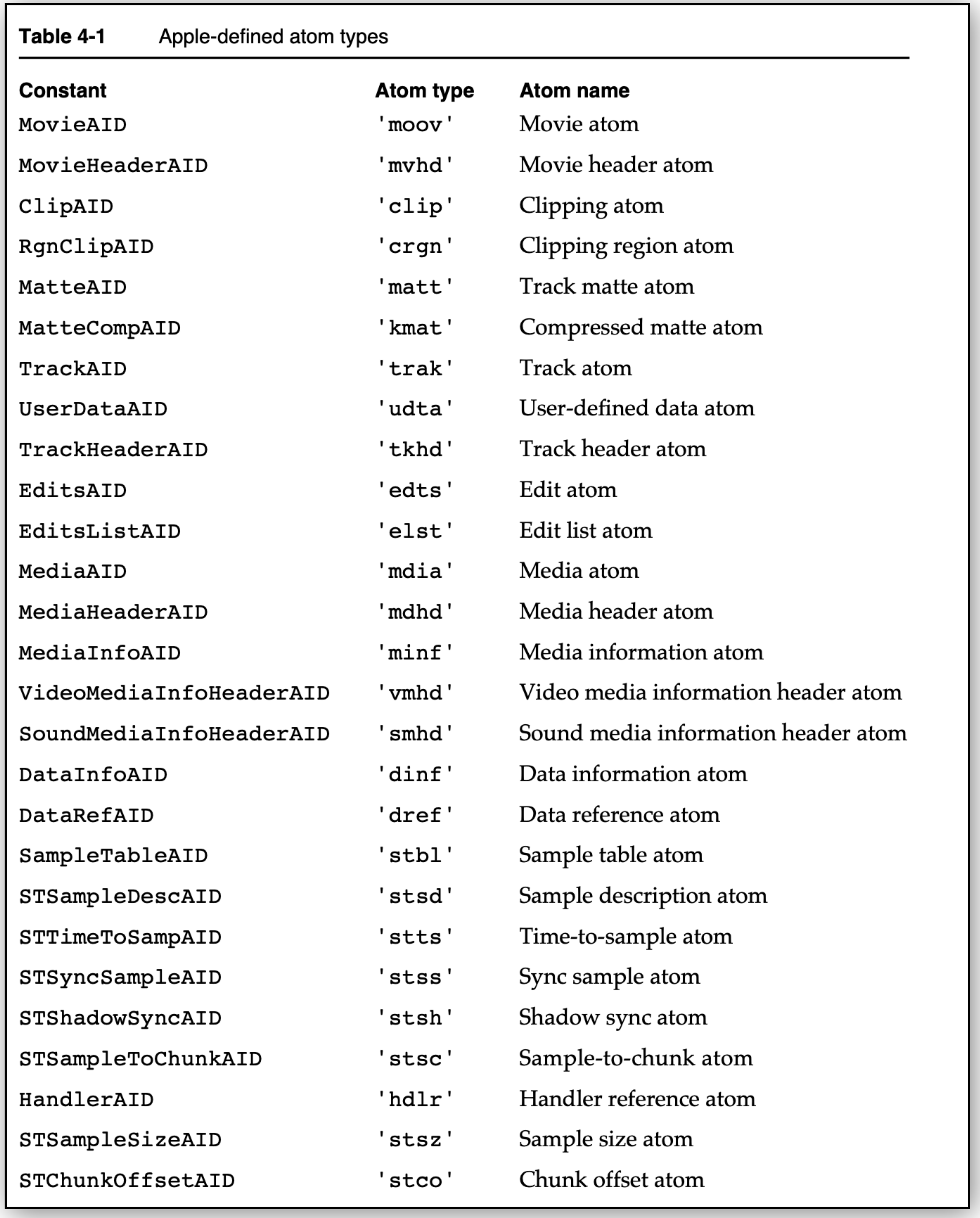
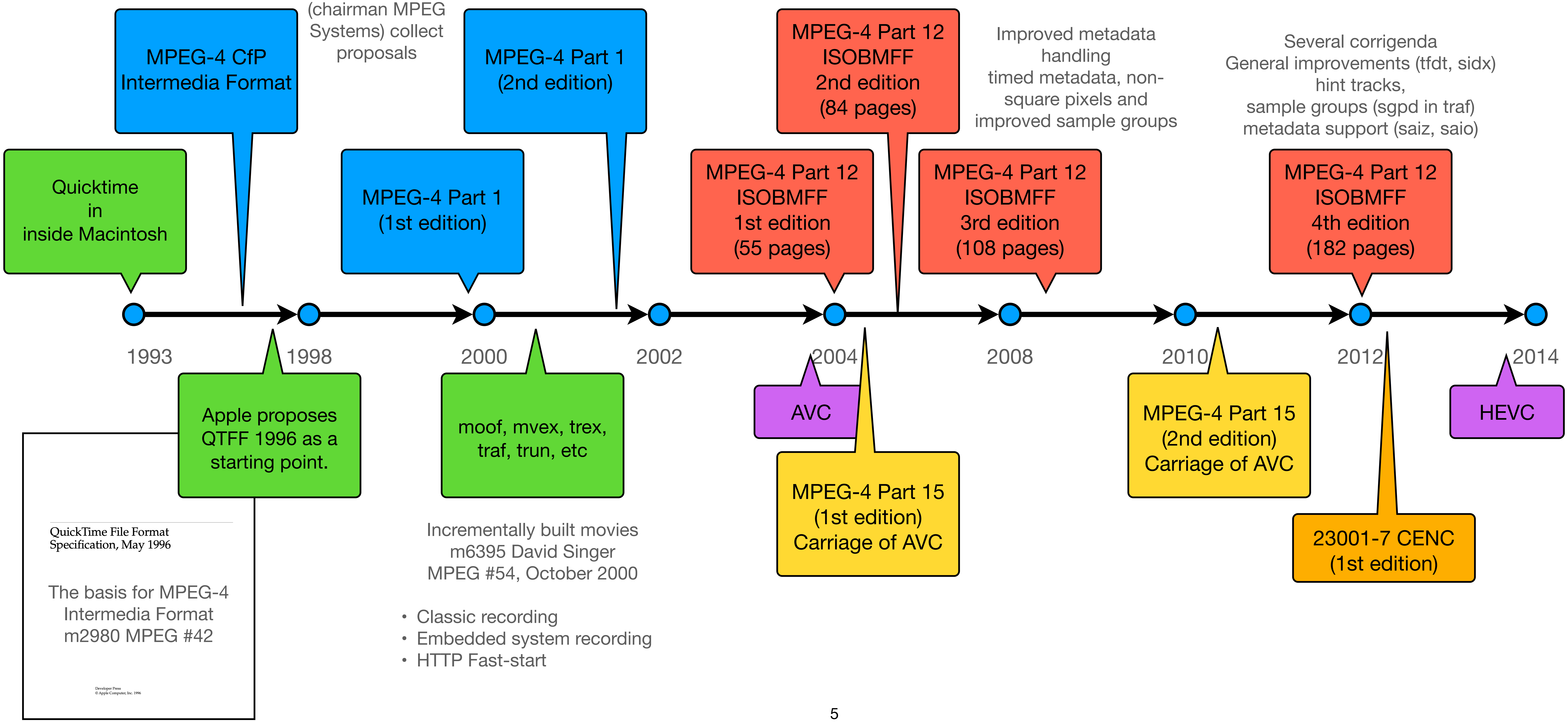


Table 4-1 Apple-defined atom types

Constant	Atom type	Atom name
MovieAID	'moov'	Movie atom
MovieHeaderAID	'mvhd'	Movie header atom
ClipAID	'clip'	Clipping atom
RgnClipAID	'crgn'	Clipping region atom
MatteAID	'matt'	Track matte atom
MatteCompAID	'kmat'	Compressed matte atom
TrackAID	'trak'	Track atom
UserDataAID	'udta'	User-defined data atom
TrackHeaderAID	'tkhd'	Track header atom
EditsAID	'edts'	Edit atom
EditsListAID	'elst'	Edit list atom
MediaAID	'mdia'	Media atom
MediaHeaderAID	'mdhd'	Media header atom
MediaInfoAID	'minf'	Media information atom
VideoMediaInfoHeaderAID	'vmhd'	Video media information header atom
SoundMediaInfoHeaderAID	'smhd'	Sound media information header atom
DataInfoAID	'dinf'	Data information atom
DataRefAID	'dref'	Data reference atom
SampleTableAID	'stbl'	Sample table atom
STSampleDescAID	'stsd'	Sample description atom
STTimeToSampAID	'stts'	Time-to-sample atom
STSyncSampleAID	'stss'	Sync sample atom
STShadowSyncAID	'stsh'	Shadow sync atom
STSampleToChunkAID	'stsc'	Sample-to-chunk atom
HandlerAID	'hdlr'	Handler reference atom
STSampleSizeAID	'stsz'	Sample size atom
STChunkOffsetAID	'stco'	Chunk offset atom

History

Part 1



Quicktime
in
inside Macintosh

MPEG-4 CfP
Intermedia Format

Rob Koenen (Chairman MPEG Requirements) and Olivier Avaro (chairman MPEG Systems) collect proposals

Apple proposes QTFF 1996 as a starting point.

QuickTime File Format
Specification, May 1996

The basis for MPEG-4 Intermedia Format
m2980 MPEG #42

Developer Press
© Apple Computer, Inc. 1996

MPEG-4 Part 1
(1st edition)

moof, mvex, trex, traf, trun, etc

Incrementally built movies
m6395 David Singer
MPEG #54, October 2000

- Classic recording
- Embedded system recording
- HTTP Fast-start

MPEG-4 Part 1
(2nd edition)

MPEG-4 Part 12
ISOBMFF
1st edition
(55 pages)

AVC

MPEG-4 Part 15
(1st edition)
Carriage of AVC

new tools,
(Fragmented movie support?)
content protection, better support of metadata, better support for advanced coding

MPEG-4 Part 12
ISOBMFF
2nd edition
(84 pages)

Improved metadata handling
timed metadata, non-square pixels and improved sample groups

MPEG-4 Part 12
ISOBMFF
3rd edition
(108 pages)

MPEG-4 Part 15
(2nd edition)
Carriage of AVC

Several corrigenda
General improvements (tfdt, sidx)
hint tracks,
sample groups (sgpd in traf)
metadata support (saiz, saio)

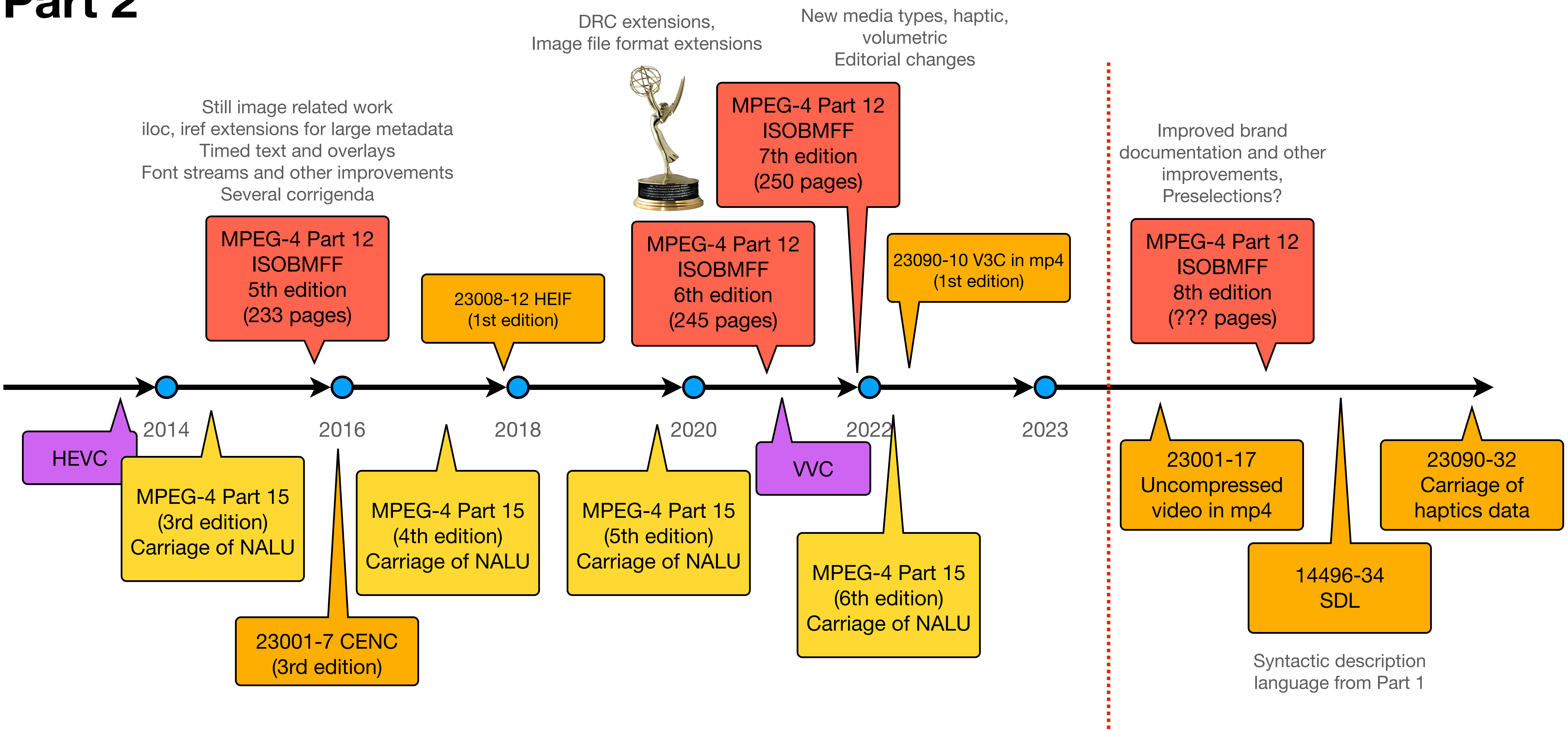
MPEG-4 Part 12
ISOBMFF
4th edition
(182 pages)

23001-7 CENC
(1st edition)

HEVC

History

Part 2



HEVC

2014

MPEG-4 Part 15 (3rd edition) Carriage of NALU

Still image related work
iloc, iref extensions for large metadata
Timed text and overlays
Font streams and other improvements
Several corrigenda

MPEG-4 Part 12 ISOBMFF 5th edition (233 pages)

2016

23001-7 CENC (3rd edition)

MPEG-4 Part 15 (4th edition) Carriage of NALU

23008-12 HEIF (1st edition)

2018

DRC extensions,
Image file format extensions

MPEG-4 Part 12 ISOBMFF 6th edition (245 pages)

MPEG-4 Part 15 (5th edition) Carriage of NALU

2020

VVC

New media types, haptic, volumetric
Editorial changes

MPEG-4 Part 12 ISOBMFF 7th edition (250 pages)

2022

MPEG-4 Part 15 (6th edition) Carriage of NALU

23090-10 V3C in mp4 (1st edition)

2023

Improved brand documentation and other improvements,
Preselections?

MPEG-4 Part 12 ISOBMFF 8th edition (??? pages)

23001-17 Uncompressed video in mp4

14496-34 SDL

Syntactic description language from Part 1

23090-32 Carriage of haptics data

2020 Technology and Engineering Achievement Emmy Award Winners

Standardization of the ISO Base Media File Format

File Format Subgroup
under ISO/IEC JTC1/SC29/WG 3

- Previous Chair: David Singer (Apple)
- Current Chair: Cyril Concolato (Netflix)
- Current Co-Chair: Miska Hannuksela (Nokia)

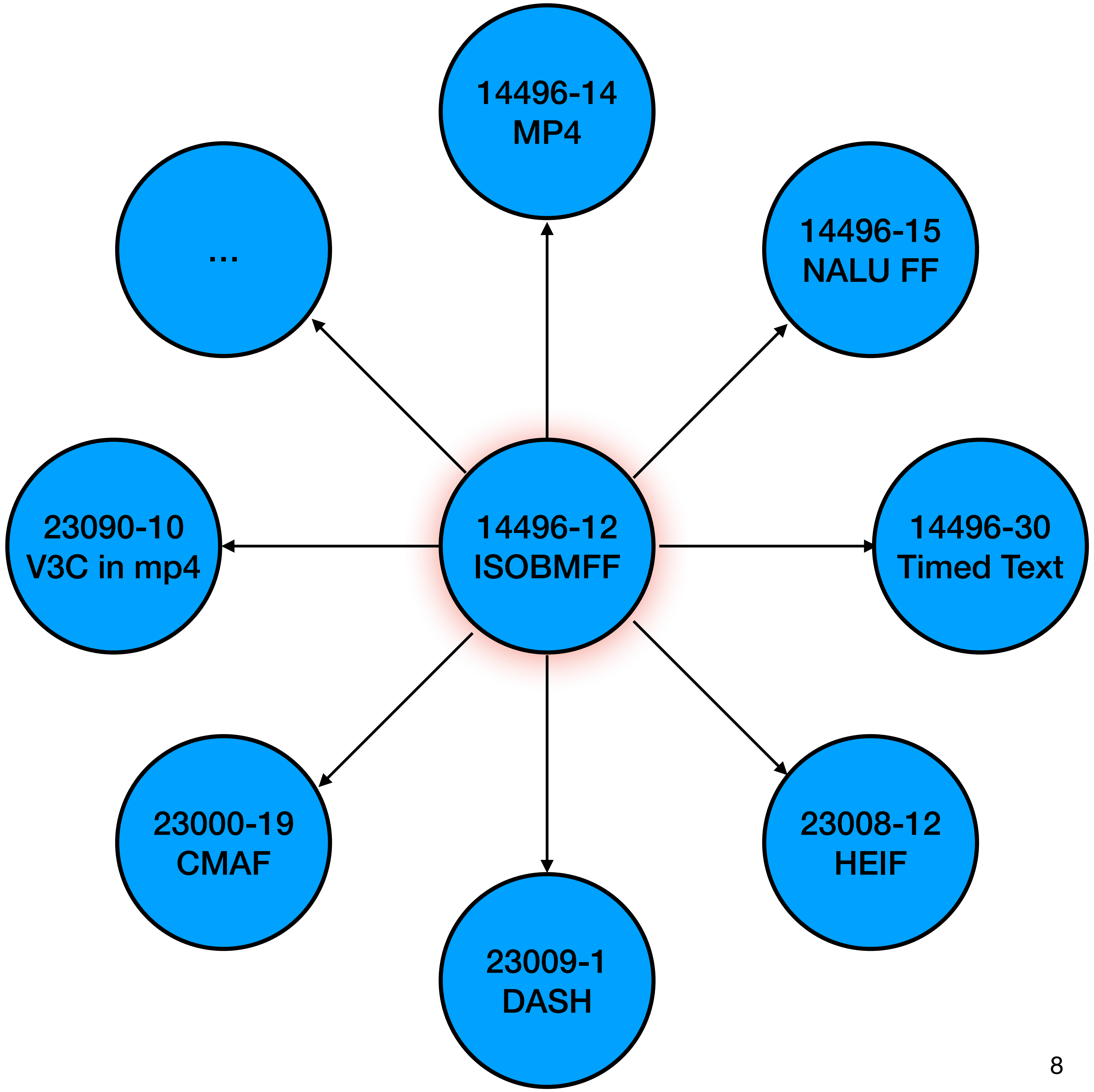
https://theemmys.tv/tech-72nd-award-recipients

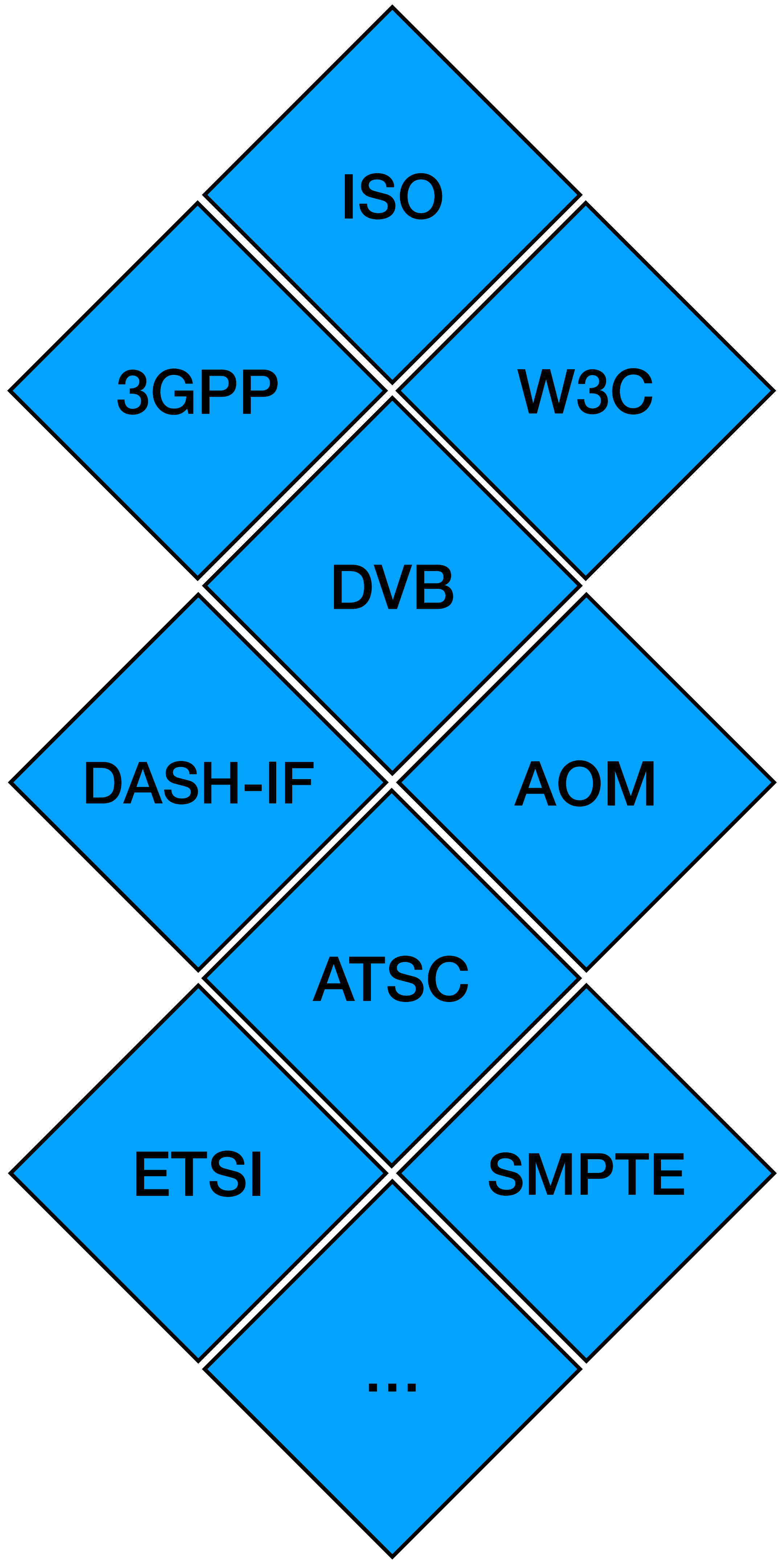
David Singer

Cyril Concolato

Solid Foundation of the File Format

Basis of many specifications in the industry



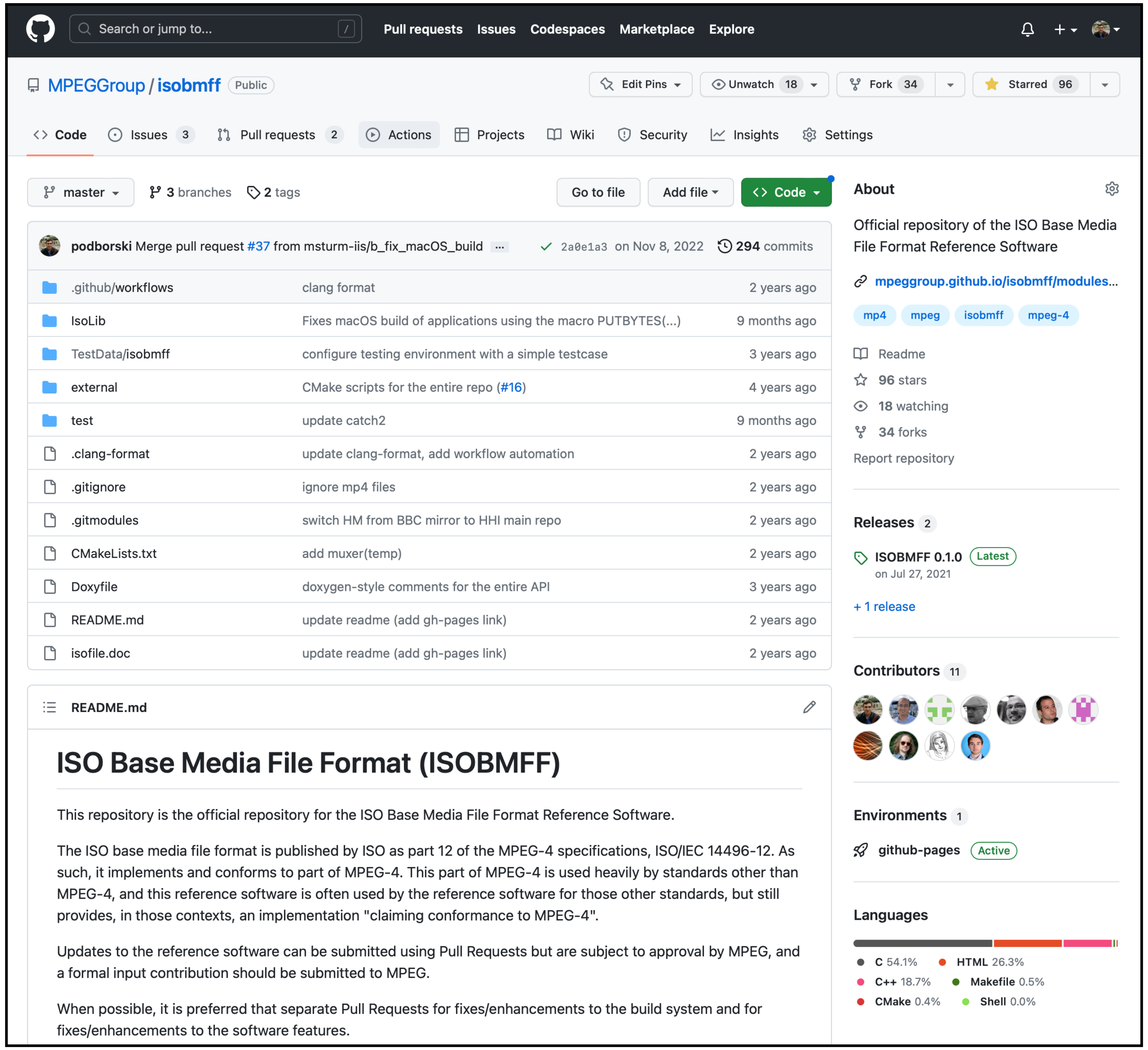


Reference Software

https://github.com/MPEGGroup/isobmff

- Every successful spec comes in a bundle with reference software & conformance suite
- SW was provided by Apple to MPEG in 90s
- Core library written in C
- Some applications in C++

Call for Proposals for an MPEG-4 Intermedia Format, w1919:
“Proposers are advised that, upon acceptance by MPEG for further evaluation, **MPEG requires that a working implementation, integrated into the existing MPEG-4 systems reference software**, be made available **before the technology can be included in the specification.** …”



podborski Merge pull request #37 from msturm-iis/b_fix_macOS_build … ✓ 2a0e1a3 on Nov 8, 2022 294 commits

.github/workflows	clang format	2 years ago
IsoLib	Fixes macOS build of applications using the macro PUTBYTES(...)	9 months ago
TestData/isobmff	configure testing environment with a simple testcase	3 years ago
external	CMake scripts for the entire repo (#16)	4 years ago
test	update catch2	9 months ago
.clang-format	update clang-format, add workflow automation	2 years ago
.gitignore	ignore mp4 files	2 years ago
.gitmodules	switch HM from BBC mirror to HHI main repo	2 years ago
CMakeLists.txt	add muxer(temp)	2 years ago
Doxyfile	doxygen-style comments for the entire API	3 years ago
README.md	update readme (add gh-pages link)	2 years ago
isofile.doc	update readme (add gh-pages link)	2 years ago

README.md

ISO Base Media File Format (ISOBMFF)

This repository is the official repository for the ISO Base Media File Format Reference Software.

The ISO base media file format is published by ISO as part 12 of the MPEG-4 specifications, ISO/IEC 14496-12. As such, it implements and conforms to part of MPEG-4. This part of MPEG-4 is used heavily by standards other than MPEG-4, and this reference software is often used by the reference software for those other standards, but still provides, in those contexts, an implementation "claiming conformance to MPEG-4".

Updates to the reference software can be submitted using Pull Requests but are subject to approval by MPEG, and a formal input contribution should be submitted to MPEG.

When possible, it is preferred that separate Pull Requests for fixes/enhancements to the build system and for fixes/enhancements to the software features.

About

Official repository of the ISO Base Media File Format Reference Software

mpeggroup.github.io/isobmff/modules...

mp4 mpeg isobmff mpeg-4

Readme
96 stars
18 watching
34 forks
Report repository

Releases 2

ISOBMFF 0.1.0 Latest
on Jul 27, 2021

+ 1 release

Contributors 11

Environments 1

github-pages Active

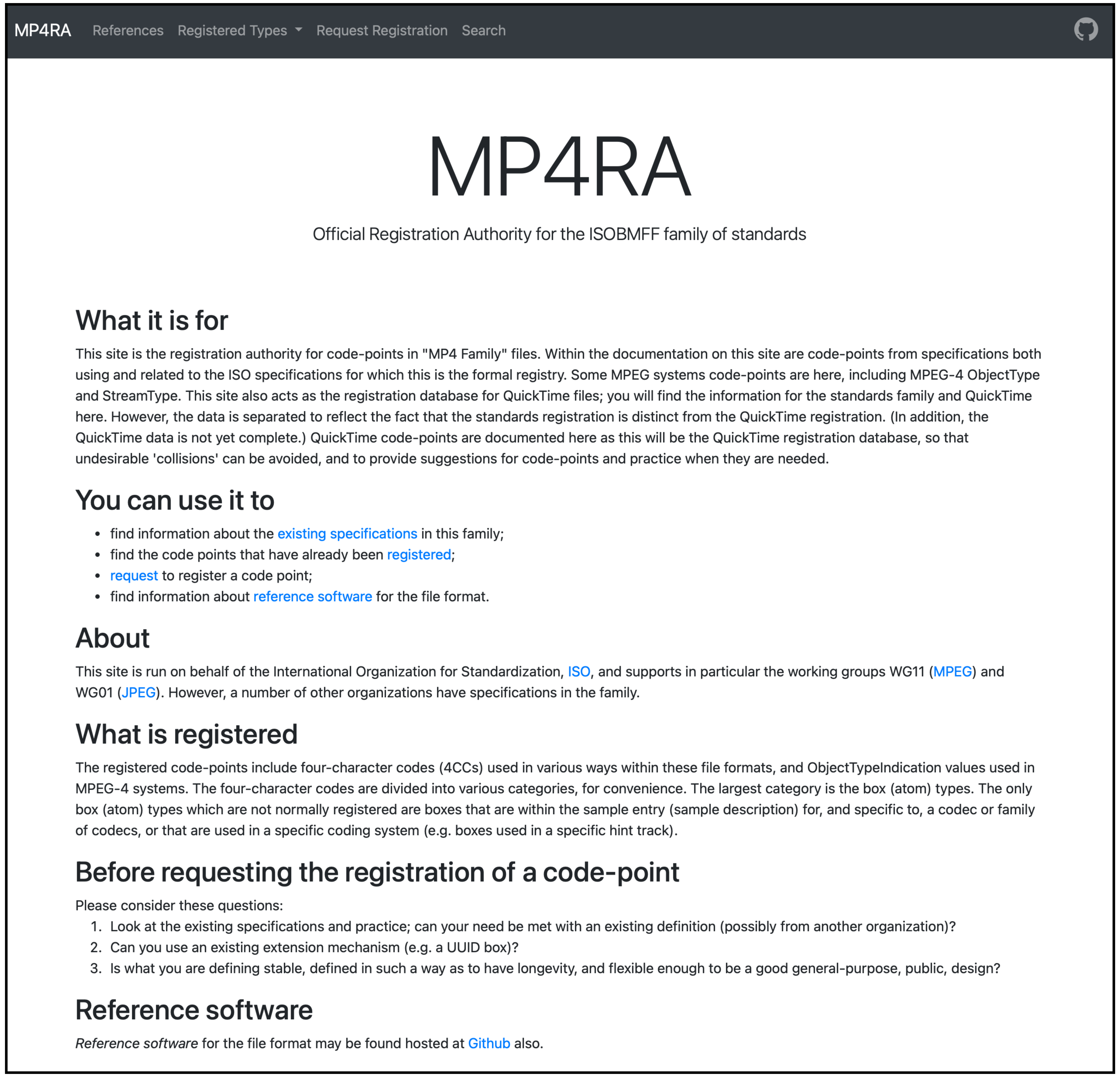
Languages

● C 54.1% ● HTML 26.3%
● C++ 18.7% ● Makefile 0.5%
● CMake 0.4% ● Shell 0.0%

MP4RA

Registration Authority for 4CCs

- http://mp4ra.org
- Registers code-points and documents in the file-format family of standards
- Avoid collisions
- Publicly available
- Simple
- Free



MP4RA

Official Registration Authority for the ISOBMFF family of standards

What it is for

This site is the registration authority for code-points in "MP4 Family" files. Within the documentation on this site are code-points from specifications both using and related to the ISO specifications for which this is the formal registry. Some MPEG systems code-points are here, including MPEG-4 ObjectType and StreamType. This site also acts as the registration database for QuickTime files; you will find the information for the standards family and QuickTime here. However, the data is separated to reflect the fact that the standards registration is distinct from the QuickTime registration. (In addition, the QuickTime data is not yet complete.) QuickTime code-points are documented here as this will be the QuickTime registration database, so that undesirable 'collisions' can be avoided, and to provide suggestions for code-points and practice when they are needed.

You can use it to

- find information about the existing specifications in this family;
- find the code points that have already been registered;
- request to register a code point;
- find information about reference software for the file format.

About

This site is run on behalf of the International Organization for Standardization, ISO, and supports in particular the working groups WG11 (MPEG) and WG01 (JPEG). However, a number of other organizations have specifications in the family.

What is registered

The registered code-points include four-character codes (4CCs) used in various ways within these file formats, and ObjectTypeIndication values used in MPEG-4 systems. The four-character codes are divided into various categories, for convenience. The largest category is the box (atom) types. The only box (atom) types which are not normally registered are boxes that are within the sample entry (sample description) for, and specific to, a codec or family of codecs, or that are used in a specific coding system (e.g. boxes used in a specific hint track).

Before requesting the registration of a code-point

Please consider these questions:

1. Look at the existing specifications and practice; can your need be met with an existing definition (possibly from another organization)?
2. Can you use an existing extension mechanism (e.g. a UUID box)?
3. Is what you are defining stable, defined in such a way as to have longevity, and flexible enough to be a good general-purpose, public, design?

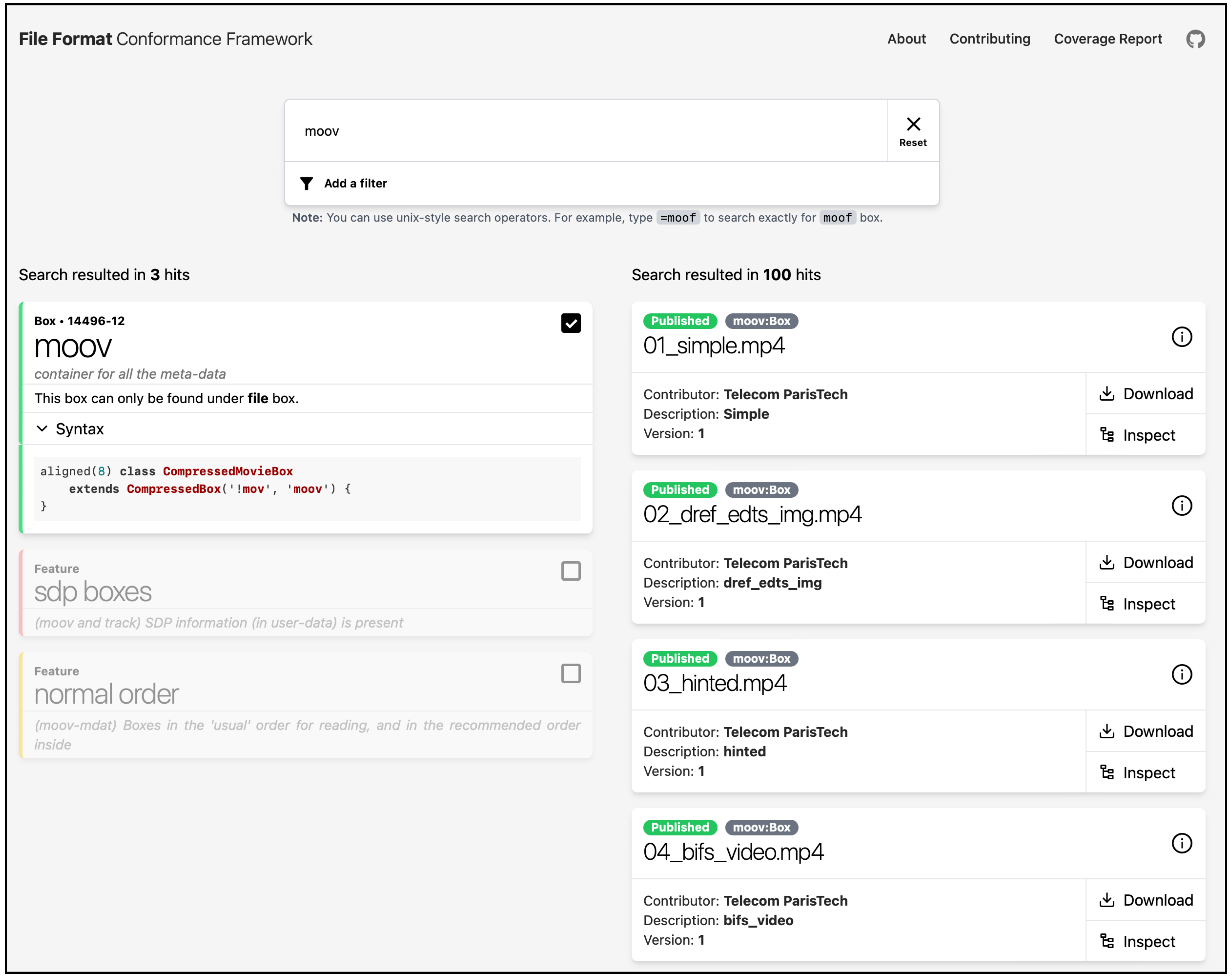
Reference software

Reference software for the file format may be found hosted at Github also.

Conformance Suite

https://github.com/MPEGGroup/FileFormatConformance

- ISO/IEC 14496-32
- Conformance files are now mandatory for new features to be accepted in the spec.
- We are currently re-designing the conformance suite (WIP)
- Git-based approach
- Automation
    - Conformance file submissions
    - Coverage statistics
    - Search by feature



File Format Conformance Framework

Note: You can use unix-style search operators. For example, type `=moof` to search exactly for `moof` box.

Search resulted in **3** hits

Box • 14496-12 ☑

moov

container for all the meta-data

This box can only be found under **file** box.

Syntax

```
aligned(8) class CompressedMovieBox
    extends CompressedBox('!mov', 'moov') {
}
```

Feature ☐

sdp boxes

(moov and track) SDP information (in user-data) is present

Feature ☐

normal order

(moov-mdat) Boxes in the 'usual' order for reading, and in the recommended order inside

Search resulted in **100** hits

Published moov:Box

01_simple.mp4

Contributor: **Telecom ParisTech**
Description: **Simple**
Version: **1**

Published moov:Box

02_dref_edts_img.mp4

Contributor: **Telecom ParisTech**
Description: **dref_edts_img**
Version: **1**

Published moov:Box

03_hinted.mp4

Contributor: **Telecom ParisTech**
Description: **hinted**
Version: **1**

Published moov:Box

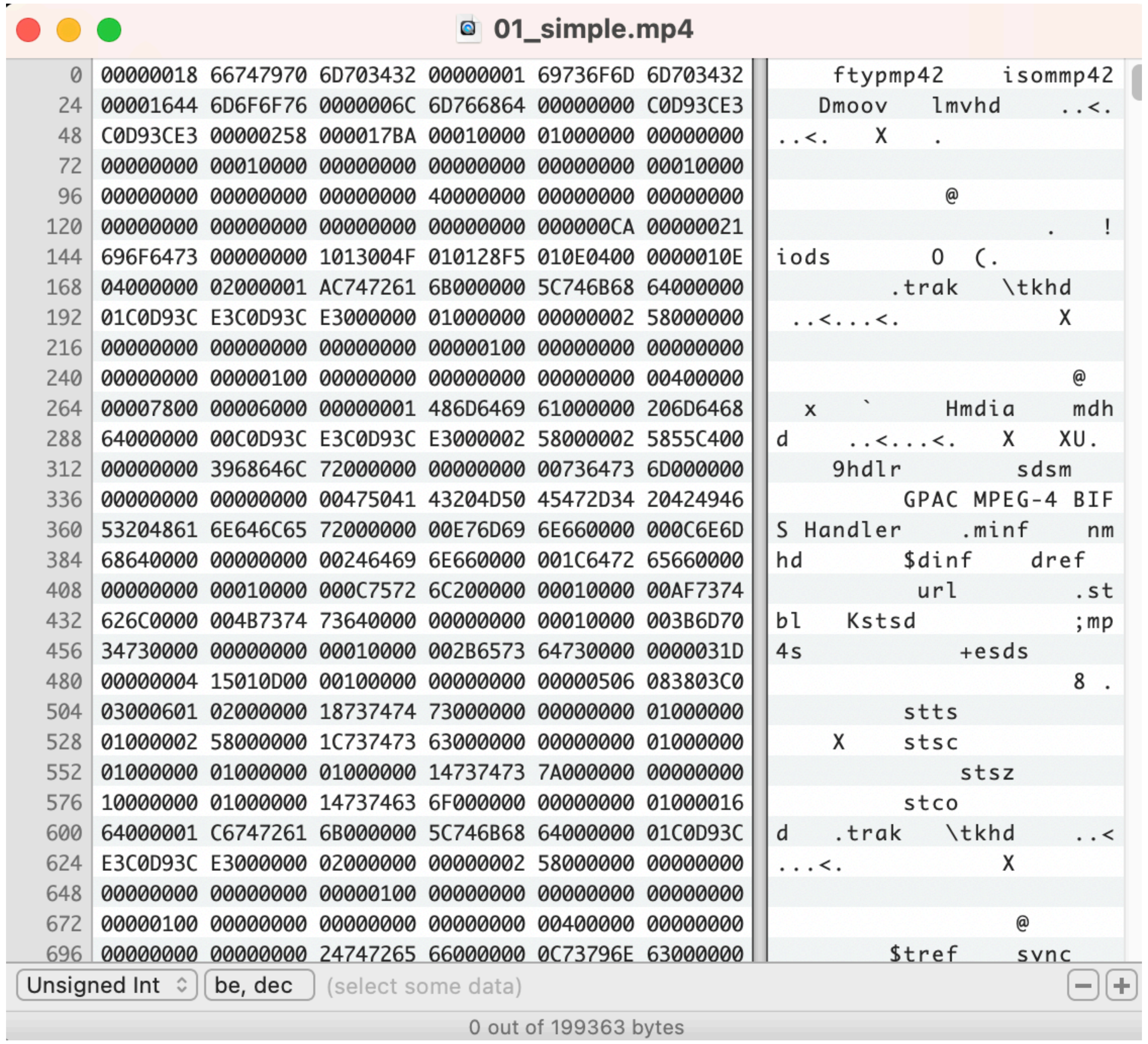
04_bifs_video.mp4

Contributor: **Telecom ParisTech**
Description: **bifs_video**
Version: **1**

Other tools

Huge selection of tools, but none implements all the capabilities of ISOBMFF

- GPAC (MP4Box has perhaps the most coverage today)
- FFmpeg
- Bento4
- Mediainfo
- A lot of tools on GitHub
- Hex editors, if you are really into it 😉

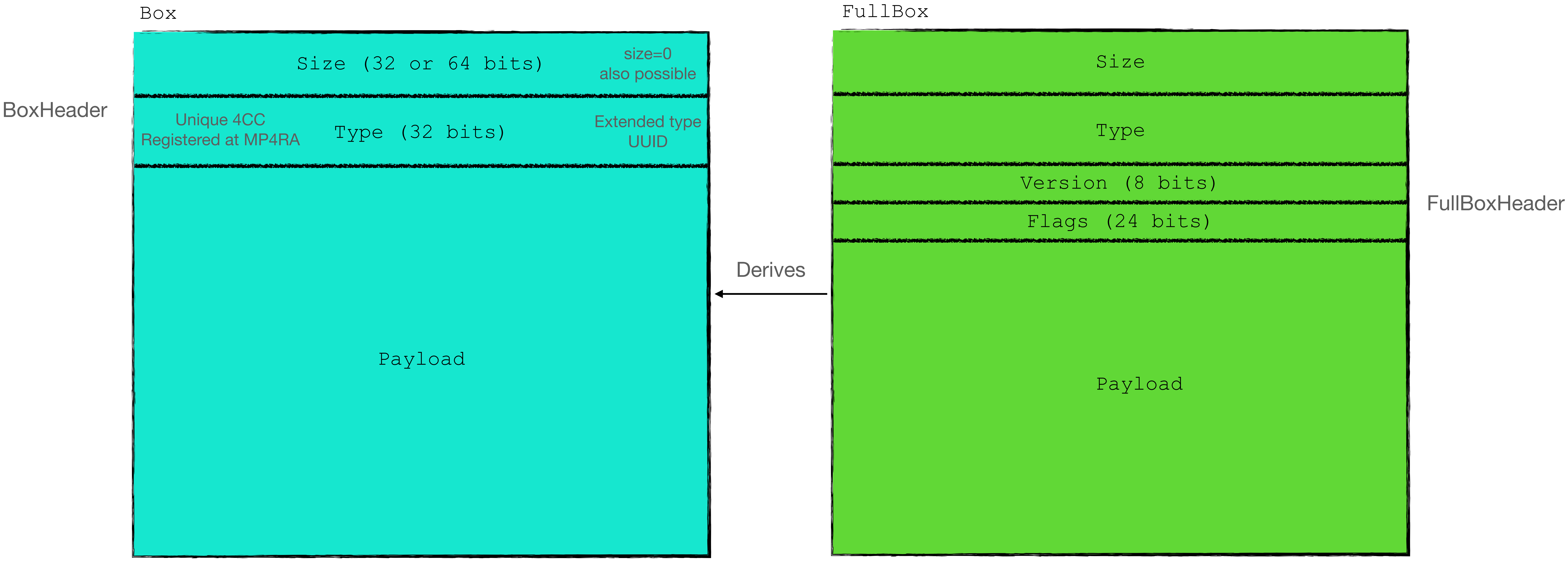


```
  0 00000018 66747970 6D703432 00000001 69736F6D 6D703432     ftypmp42     isommp42
 24 00001644 6D6F6F76 0000006C 6D766864 00000000 C0D93CE3    Dmoov   lmvhd      ..<.
 48 C0D93CE3 00000258 000017BA 00010000 01000000 00000000  ..<.    X    .
 72 00000000 00010000 00000000 00000000 00000000 00010000
 96 00000000 00000000 00000000 40000000 00000000 00000000              @
120 00000000 00000000 00000000 00000000 000000CA 00000021                        .   !
144 696F6473 00000000 1013004F 010128F5 010E0400 0000010E  iods         O  (.
168 04000000 02000001 AC747261 6B000000 5C746B68 64000000            .trak   \tkhd
192 01C0D93C E3C0D93C E3000000 01000000 00000002 58000000   ..<...<.               X
216 00000000 00000000 00000000 00000100 00000000 00000000
240 00000000 00000100 00000000 00000000 00000000 00400000                          @
264 00007800 00006000 00000001 486D6469 61000000 206D6468    x    `       Hmdia    mdh
288 64000000 00C0D93C E3C0D93C E3000002 58000002 5855C400  d     ..<...<.    X    XU.
312 00000000 3968646C 72000000 00000000 00736473 6D000000      9hdlr          sdsm
336 00000000 00000000 00475041 43204D50 45472D34 20424946              GPAC MPEG-4 BIF
360 53204861 6E646C65 72000000 00E76D69 6E660000 000C6E6D  S Handler      .minf     nm
384 68640000 00000000 00246469 6E660000 001C6472 65660000  hd            $dinf     dref
408 00000000 00010000 000C7572 6C200000 00010000 00AF7374                url          .st
432 626C0000 004B7374 73640000 00000000 00010000 003B6D70  bl    Kstsd               ;mp
456 34730000 00000000 00010000 002B6573 64730000 0000031D  4s                +esds
480 00000004 15010D00 00100000 00000000 00000506 083803C0                           8 .
504 03000601 02000000 18737474 73000000 00000000 01000000              stts
528 01000002 58000000 1C737473 63000000 00000000 01000000       X      stsc
552 01000000 01000000 01000000 14737473 7A000000 00000000                   stsz
576 10000000 01000000 14737463 6F000000 00000000 01000016              stco
600 64000001 C6747261 6B000000 5C746B68 64000000 01C0D93C  d    .trak    \tkhd      ..<
624 E3C0D93C E3000000 02000000 00000002 58000000 00000000  ...<.                X
648 00000000 00000000 00000100 00000000 00000000 00000000
672 00000100 00000000 00000000 00000000 00400000 00000000                    @
696 00000000 00000000 24747265 66000000 0C73796E 63000000            $tref     sync
```

0 out of 199363 bytes

ISOBMFF

Basic concepts refresher



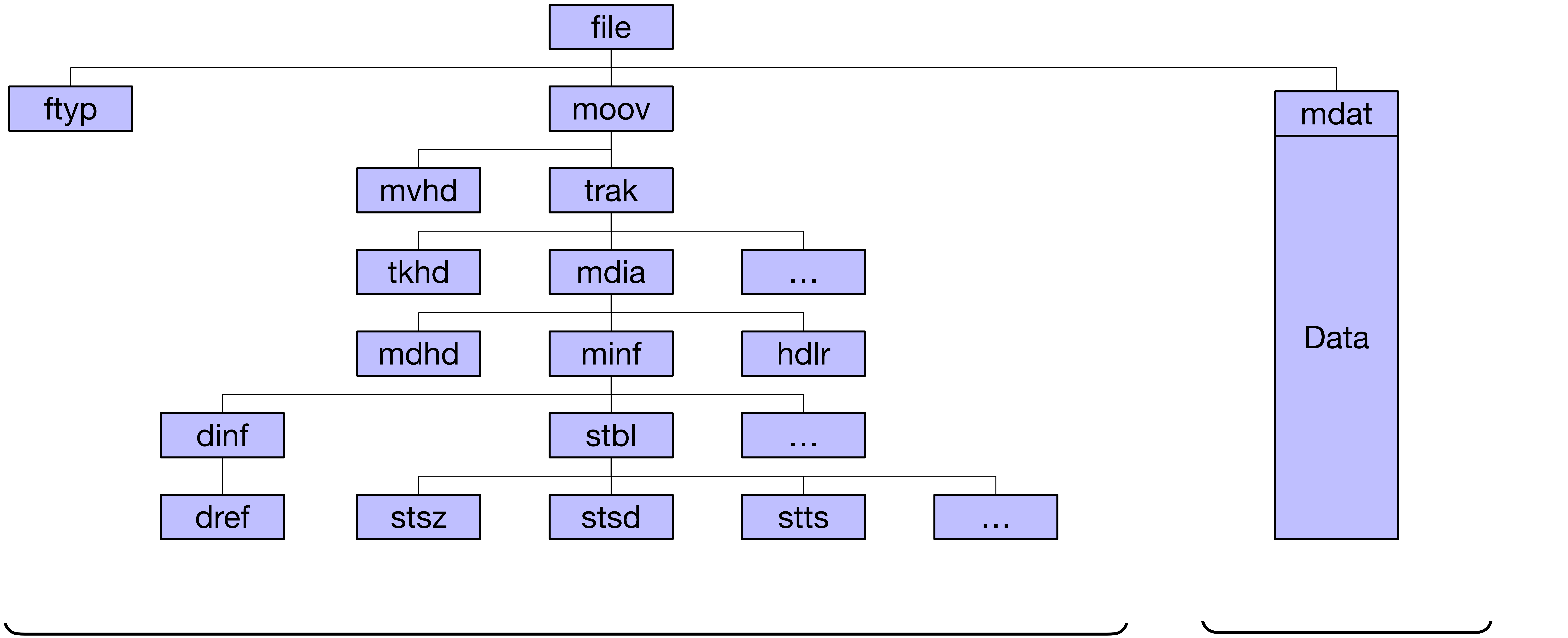
ISOBMFF

Basic concepts refresher

Movie Box - moov
Media Data Box - mdta
ftyp
* Meta Box related - meta
Item 1
Item 2
Item References
Track References
Track 1
Meta
Sample Entry 1
1
2
3
4
5
6
…
N
Track 2
Meta
Sample Entry 1
1
2
3
4
5
6
…
N
Audio / Video / Text / Metadata
…
Entity Groups
1
4
9
…
N
Data
Sample Groups (mapping)
Sample Group Description Entry
Track 1
and
Track 2
Track Groups
Track 1
or
Track 2
Alternate Groups
And much much more

Typical ISO File Box Hierarchy

Single track

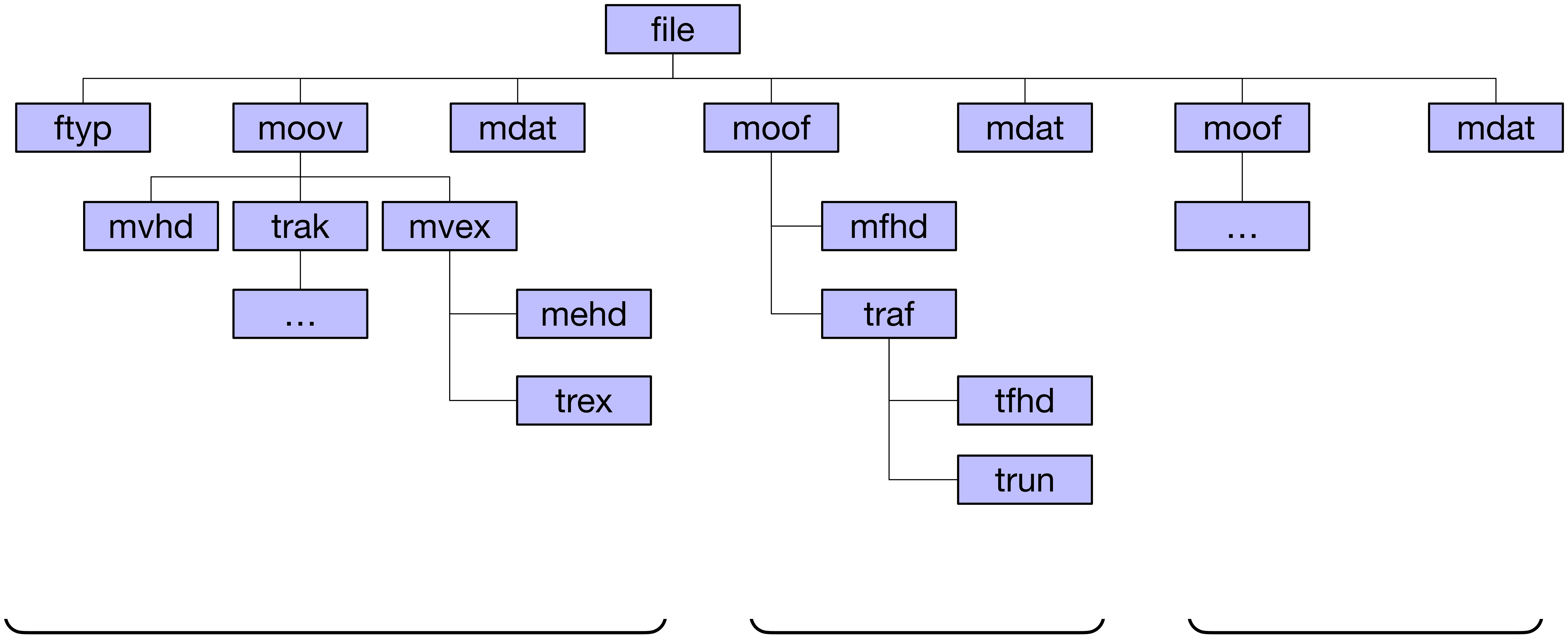


Metadata
(mostly)

Data
(Not boxed structured, mostly)

Typical ISO File Box Hierarchy

Single track, fragmented file



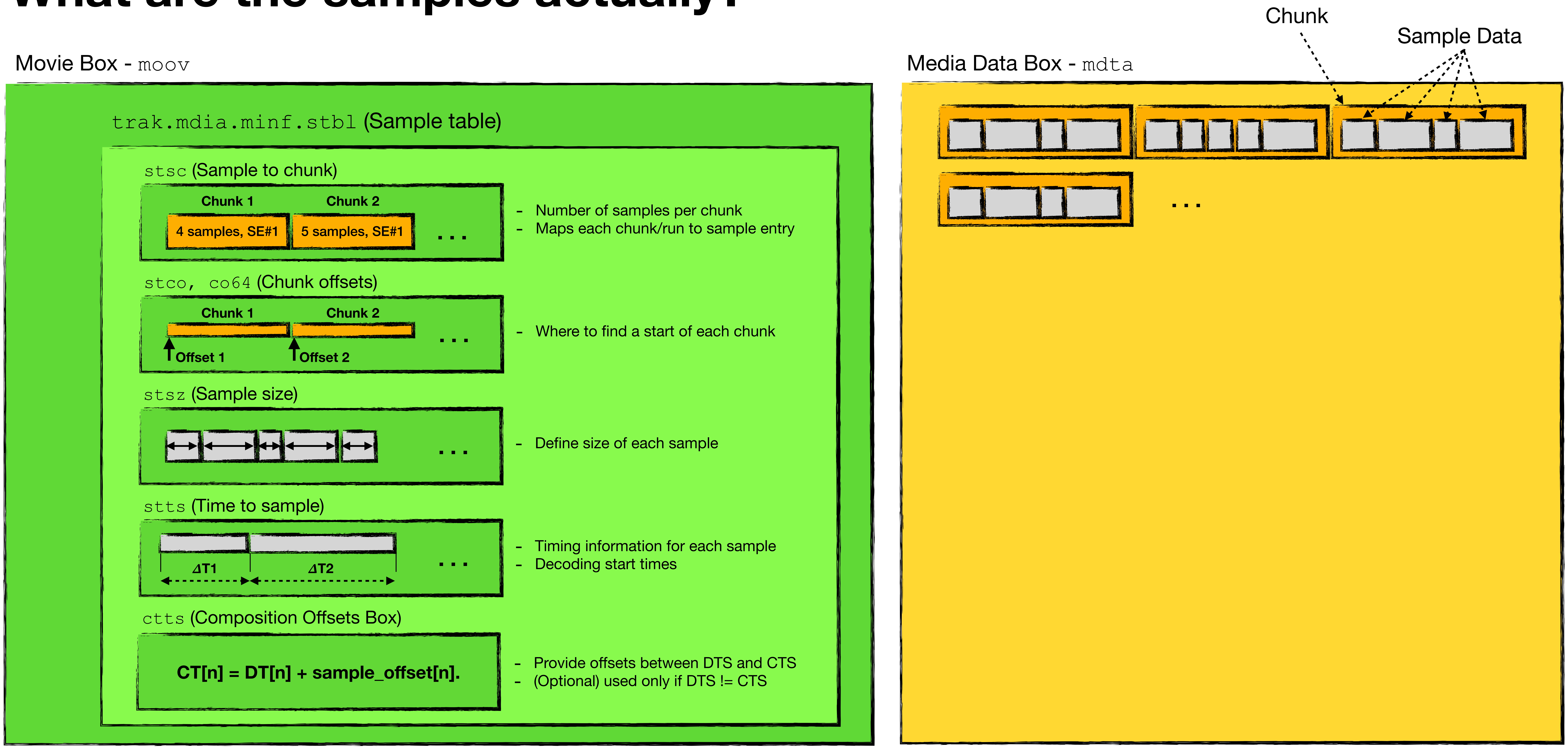
Initial movie with the indication that the file is fragmented

Fragment 1

Fragment 2

What is inside the sample table

What are the samples actually?



Movie Box - `moov`

`trak.mdia.minf.stbl` (Sample table)

`stsc` (Sample to chunk)

Chunk 1 | Chunk 2
4 samples, SE#1 | 5 samples, SE#1 . . .

- Number of samples per chunk
- Maps each chunk/run to sample entry

`stco, co64` (Chunk offsets)

Chunk 1 | Chunk 2 . . .
Offset 1 | Offset 2

- Where to find a start of each chunk

`stsz` (Sample size)

. . .

- Define size of each sample

`stts` (Time to sample)

ΔT1 | ΔT2 . . .

- Timing information for each sample
- Decoding start times

`ctts` (Composition Offsets Box)

CT[n] = DT[n] + sample_offset[n].

- Provide offsets between DTS and CTS
- (Optional) used only if DTS != CTS

Media Data Box - `mdta`

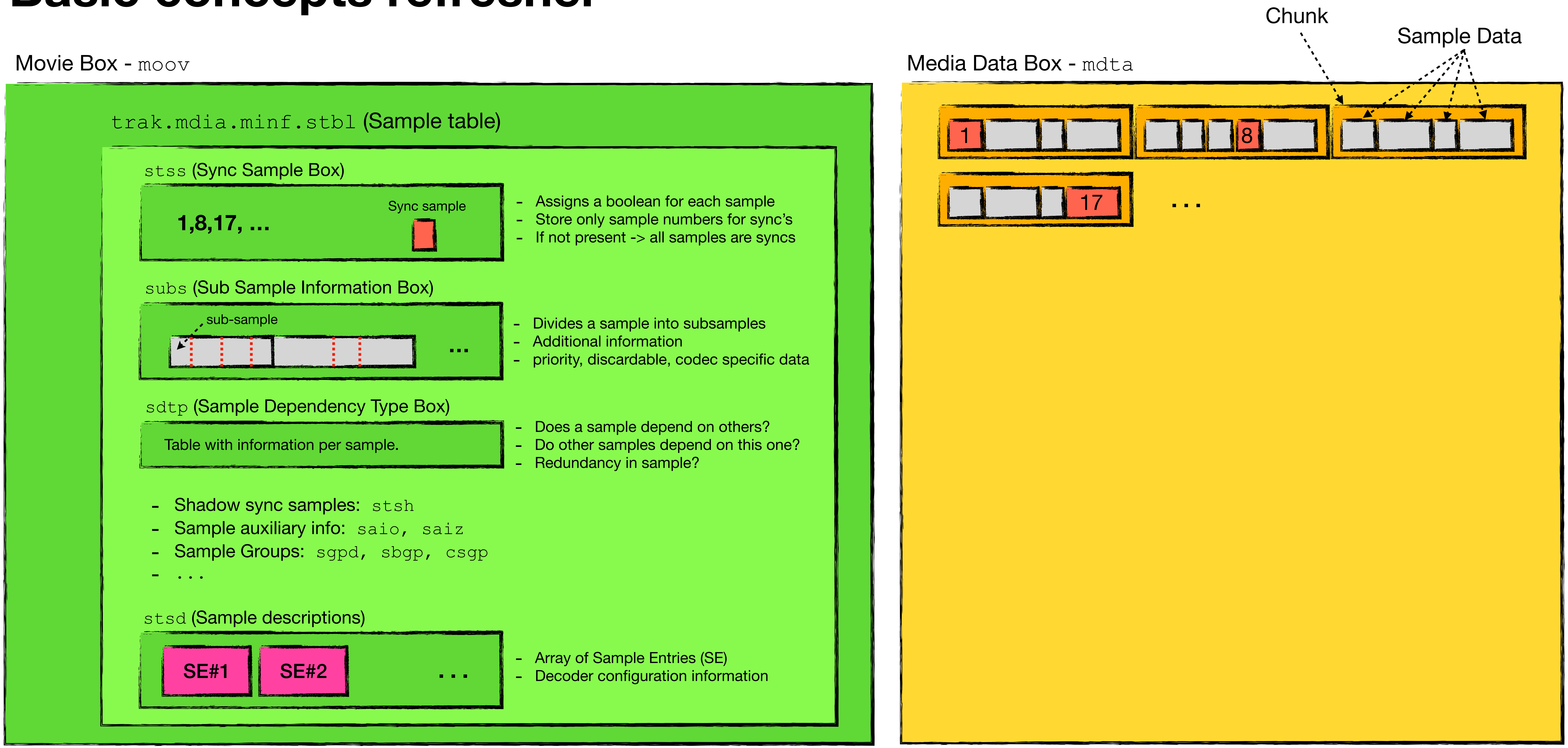
Chunk

Sample Data

. . .

What is inside the sample table

Basic concepts refresher



Movie Box - `moov`

`trak.mdia.minf.stbl` (Sample table)

`stss` (Sync Sample Box)

- Assigns a boolean for each sample
- Store only sample numbers for sync's
- If not present -> all samples are syncs

`subs` (Sub Sample Information Box)

- Divides a sample into subsamples
- Additional information
- priority, discardable, codec specific data

`sdtp` (Sample Dependency Type Box)

Table with information per sample.

- Does a sample depend on others?
- Do other samples depend on this one?
- Redundancy in sample?

- Shadow sync samples: `stsh`
- Sample auxiliary info: `saio, saiz`
- Sample Groups: `sgpd, sbgp, csgp`
- `...`

`stsd` (Sample descriptions)

- Array of Sample Entries (SE)
- Decoder configuration information

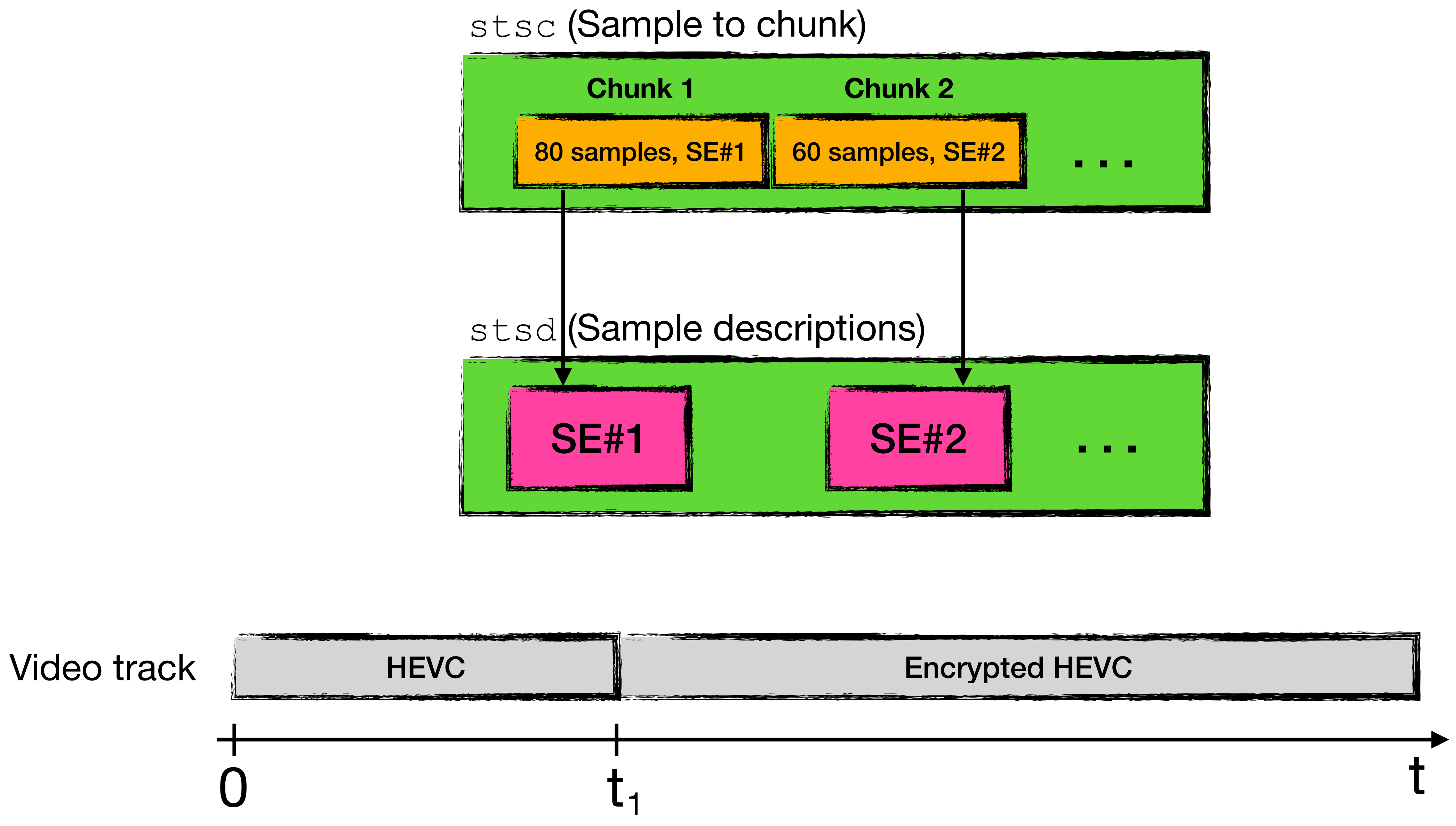
Media Data Box - `mdta`

Sample description

Sample entries, etc.

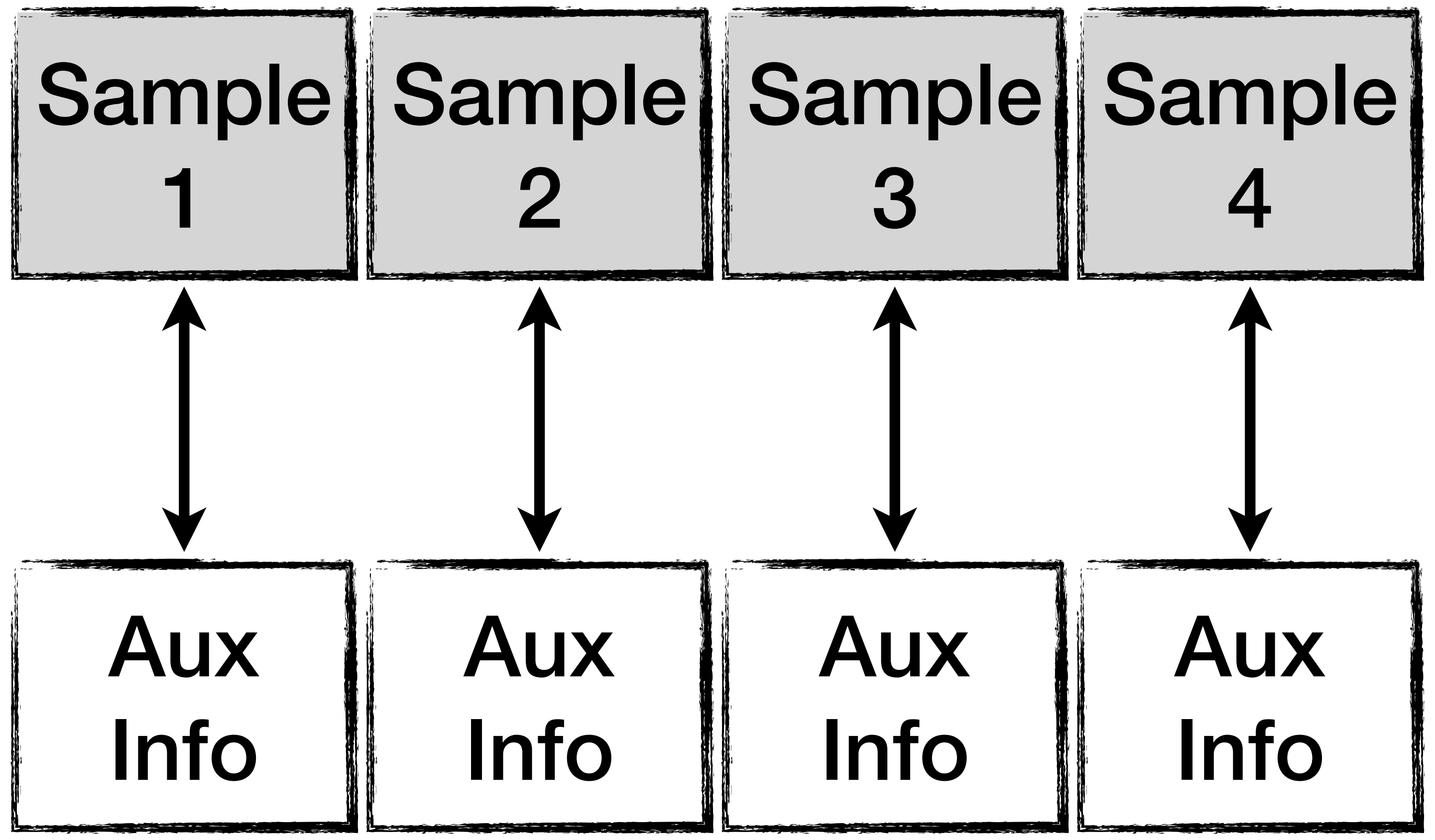
- Codec type (4CC registered at MP4RA)
  - MPEG codecs usually signal out-of-band and in-band types
  - Example: `hvc1` (out-of-band) vs `hev1` (in-band)
- Codec agnostic signaling such as:
  - Width/height, bit depth
  - Pixel aspect ratio
  - Clean Aperture
  - Color information
- Decoder configuration information.
  - Codec specific arrays of data to initialize the decoder (E.g.: parameter sets)
- Array of Sample Entries (SE)

Example for multiple sample entries



Sample auxiliary information

- Auxiliary Information is data associated to a sample, not used by the decoder, but by preprocessor or postprocessor
  ○ For example: data for decrypting the sample
- Data type indicated by a 4CC (e.g. “cenc”)
- The SampleAuxiliaryInformationSizesBox (“saiz”) indicates the additional data sizes
  - Extended in the next Amendment of ISOBMFF
- The SampleAuxiliaryInformationOffsetsBox (“saio”) indicates the additional data offsets

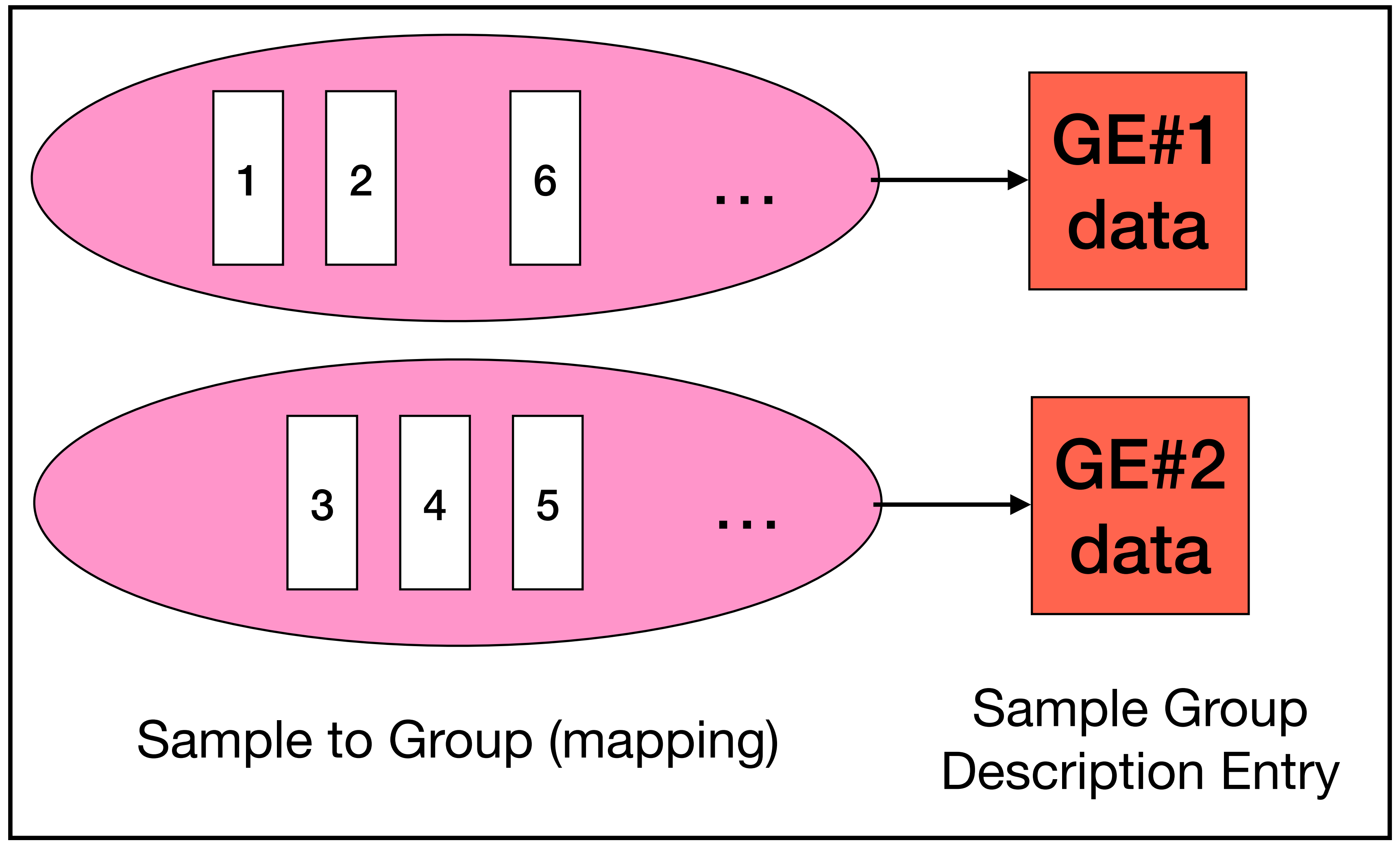


Sample groups

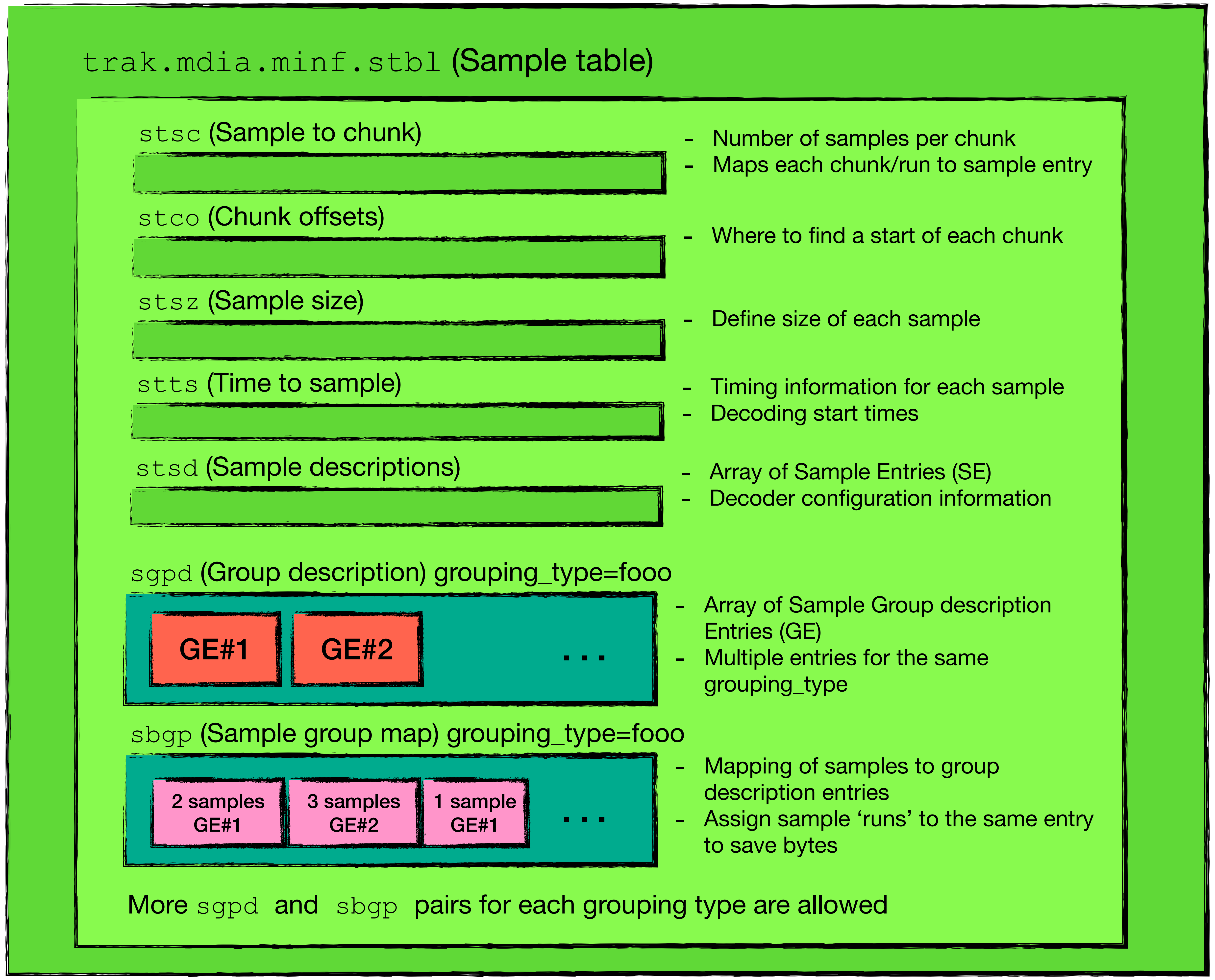
Basics of sample grouping

- Each sample group:
  - has a unique grouping type 4CC
  - Consists of:
    - **sgpd** - group description holding the data
    - **sbgp** - Sample to group mapping box
  - Each group description can have multiple description entries (derived specifications can extend the definition)
  - Sample run's are mapped to a group description entry (saving bytes)
- Multiple sample groups can be present

Example of Sample Group of grouping_type='fooo'



Movie Box - moov



trak.mdia.minf.stbl (Sample table)

stsc (Sample to chunk)
- Number of samples per chunk
- Maps each chunk/run to sample entry

stco (Chunk offsets)
- Where to find a start of each chunk

stsz (Sample size)
- Define size of each sample

stts (Time to sample)
- Timing information for each sample
- Decoding start times

stsd (Sample descriptions)
- Array of Sample Entries (SE)
- Decoder configuration information

sgpd (Group description) grouping_type=fooo
- Array of Sample Group description Entries (GE)
- Multiple entries for the same grouping_type

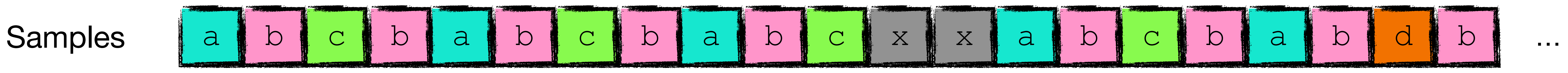
sbgp (Sample group map) grouping_type=fooo
- Mapping of samples to group description entries
- Assign sample 'runs' to the same entry to save bytes

More sgpd and sbgp pairs for each grouping type are allowed

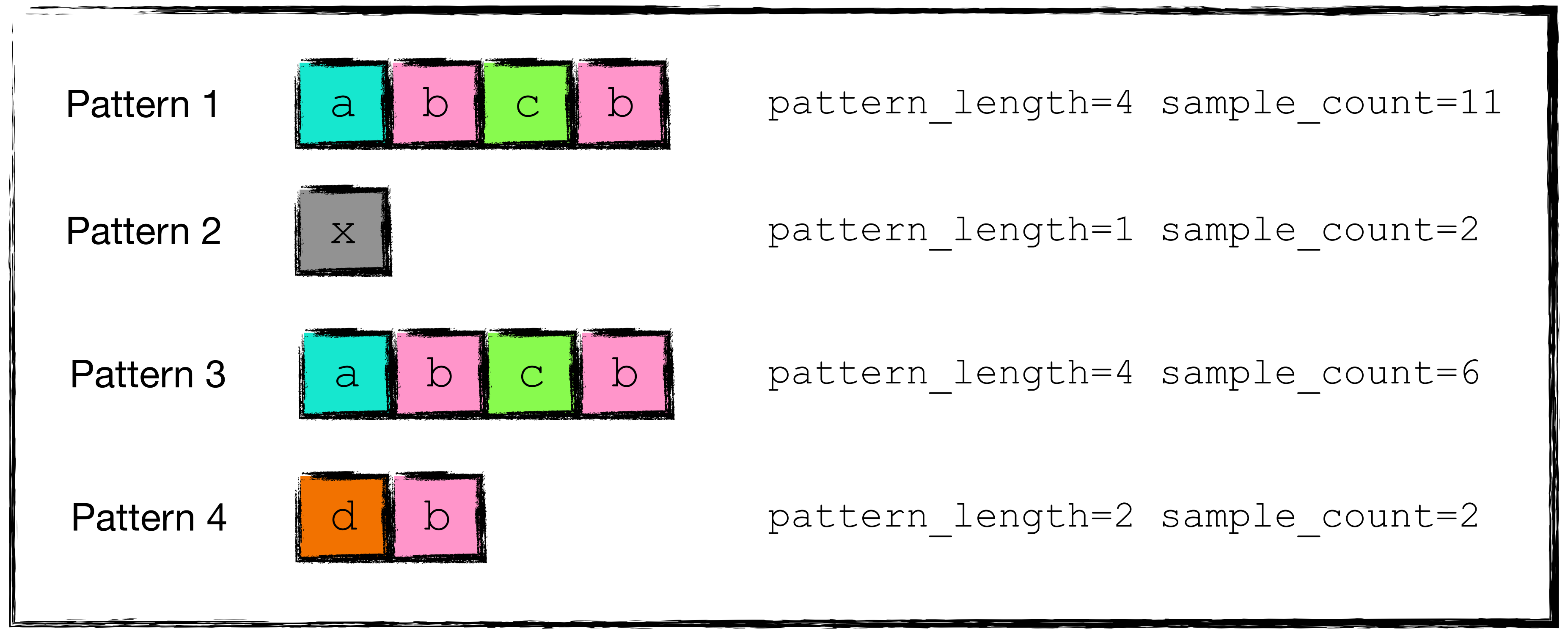
Compact sample groups

Pattern mapping

- Very often samples are mapped to group entries following a repeating pattern
- Compact sample groups are used to address this
- Replace normal mapping `sbgp` with `csgp`
- Example of compact sample to group box `csgp`:



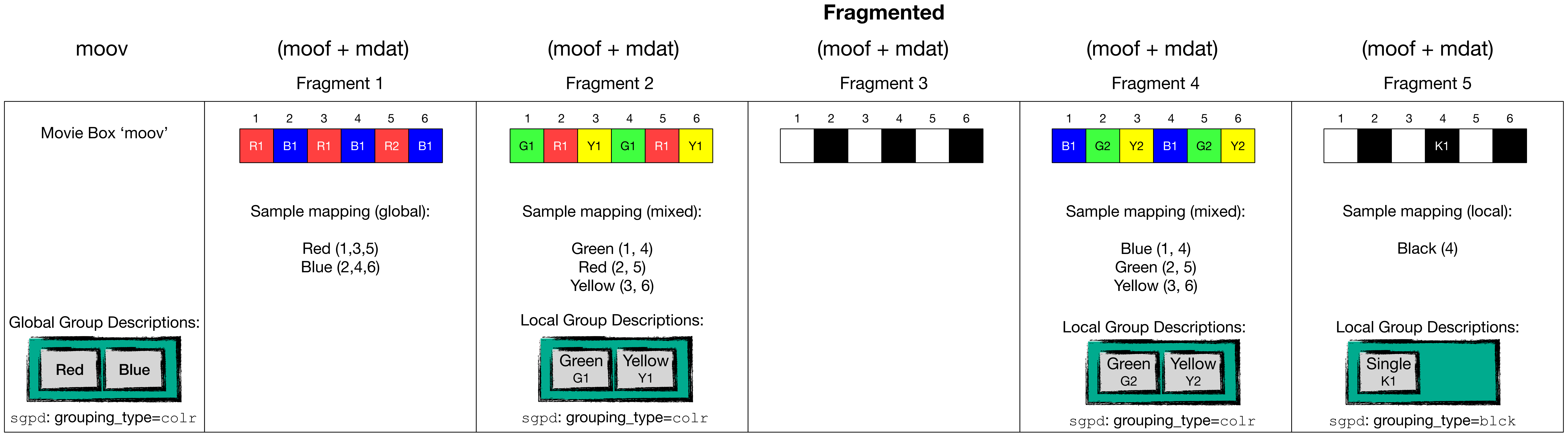
`csgp` (Compact sample to group map)



```
pattern_length=4 sample_count=11
pattern_length=1 sample_count=2
pattern_length=4 sample_count=6
pattern_length=2 sample_count=2
```

Fragmented sample groups example

Fragmented movie example



			Fragmented		
moov	(moof + mdat)	(moof + mdat)	(moof + mdat)	(moof + mdat)	(moof + mdat)
	Fragment 1	Fragment 2	Fragment 3	Fragment 4	Fragment 5
Movie Box ‘moov’	1 2 3 4 5 6 R1 B1 R1 B1 R2 B1	1 2 3 4 5 6 G1 R1 Y1 G1 R1 Y1	1 2 3 4 5 6 □ ■ □ ■ □ ■	1 2 3 4 5 6 B1 G2 Y2 B1 G2 Y2	1 2 3 4 5 6 □ ■ □ K1 □ ■
	Sample mapping (global): Red (1,3,5) Blue (2,4,6)	Sample mapping (mixed): Green (1, 4) Red (2, 5) Yellow (3, 6)		Sample mapping (mixed): Blue (1, 4) Green (2, 5) Yellow (3, 6)	Sample mapping (local): Black (4)
Global Group Descriptions: Red Blue		Local Group Descriptions: Green G1 Yellow Y1		Local Group Descriptions: Green G2 Yellow Y2	Local Group Descriptions: Single K1
`sgpd`: grouping_type=`colr`		`sgpd`: grouping_type=`colr`		`sgpd`: grouping_type=`colr`	`sgpd`: grouping_type=`blck`

- Sample groups can also be present in `traf`
- New sample groups can be added on the fly
- When defragmenting the file:
  - sample group description entries of the same type are merged into a single sample group description box
  - Local sample groups (per fragment) are added to `stbl` and mapping is updated
- Reference Software implementation

A few sample groups from ISOBMFF

More are specified in dependent specifications

- `‘alst’`: Alternative startup sequence
- `‘drap’`: Dependent random access point
- `‘prol’`: Audio pre-roll, how many samples to process until full recovery
- `‘rap ‘`: Random access point
- `‘rash’`: Rate share. Operation points for samples sharing the same rate.
- `‘roll’`: Pre/Post roll group
- `‘tele’`: Temporal Level Grouping (Scalable concepts)

Multiplexed timed metadata tracks

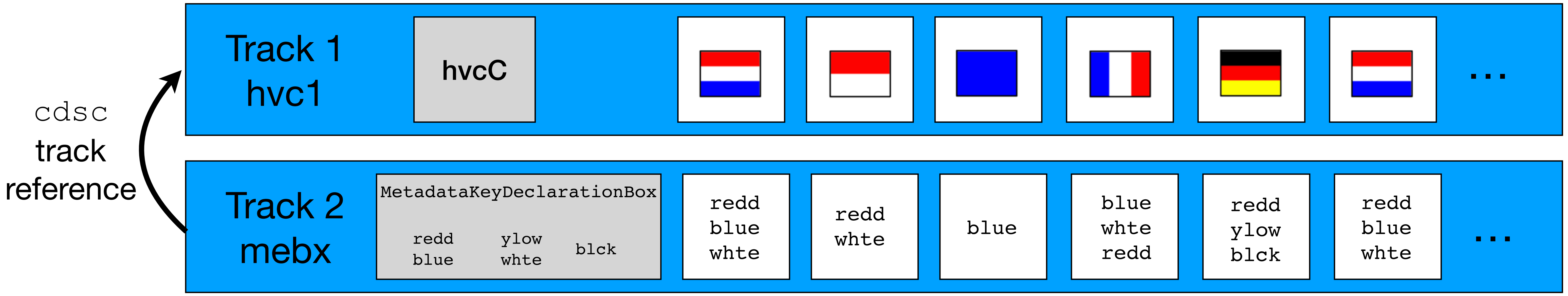
mebx - metadata boxed

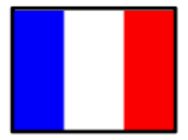
- Carriage of any user data as timed metadata synced with the “main” track
- Track references `‘cdsc’`
- Allows multiplexing different types of metadata into the same metadata sample
- Sample entry type `‘mebx’`, standing for metadata boxed
- Samples are boxes with local_key_id’s same as signaled by the `MetadataKeyDeclarationBox` in the `‘mebx’` Sample Entry

Multiplexed timed metadata tracks

Example based on color metadata and flags

mebx track example



Metadata	N Netherlands	I Indonesia	b Blue	F  France	D Germany
Red	0,0,64,16	0,0,64,24	NULL	44,0,20,48	0,16,64,16
Blue	0,32,64,16	NULL	0,0,64,48	0,0,20,48	NULL
Yellow	NULL	NULL	NULL	NULL	0,32,64,16
White	0,16,64,16	0,24,64,24	NULL	20,0,24,48	NULL
Black	NULL	NULL	NULL	NULL	0,0,64,16

Sample 1 Sample 2 Sample 3 Sample 4 Sample 5 …

Color Sample Format

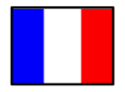
```
class ColorSample extends Box(local_key_id){
   unsigned int(32) x;
   unsigned int(32) y;
   unsigned int(32) width;
   unsigned int(32) height;
}
```

key_value from the MetadataKeyDeclarationBox is used as the local_key_id for the boxes in samples:

- redd Red color
- blue Blue color
- ylow Yellow color
- whte White color
- blck Black color

Multiplexed timed metadata tracks

Example based on color metadata and flags

Metadata	N Netherlands	I Indonesia	b Blue	F  France	D Germany
Red	0,0,64,16	0,0,64,24	NULL	44,0,20,48	0,16,64,16
Blue	0,32,64,16	NULL	0,0,64,48	0,0,20,48	NULL
Yellow	NULL	NULL	NULL	NULL	0,32,64,16
White	0,16,64,16	0,24,64,24	NULL	20,0,24,48	NULL
Black	NULL	NULL	NULL	NULL	0,0,64,16

```
Track ID 2 meta (Metadata) Enabled Self-contained
  Format meta/mebx
  Media Timescale: 30000 Duration: 30000/30000  00:00:01.000
  Num data bytes: 1728 Est. data rate: 13.824 kbps Peak data rate 0.000 kbps Nominal framerate: 30.000 fps 30 samples
  Samples:
                    [ 1] DTS     0  00:00:00.000 PTS     0  00:00:00.000 dur  1000  00:00:00.033=Offset 0x0bf4 Size
Boxed Metadata 3 items
[0]  Key(4): redd Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 10
[1]  Key(4): blue Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 20 00 00 00 40 00 00 00 10
[2]  Key(4): whte Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 10 00 00 00 40 00 00 00 10
                    [ 2] DTS  1000  00:00:00.033 PTS  1000  00:00:00.033 dur  1000  00:00:00.033 Offset 0x0c3c Size
Boxed Metadata 2 items
[0]  Key(4): redd Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 18
[1]  Key(4): whte Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 18 00 00 00 40 00 00 00 18
                    [ 3] DTS  2000  00:00:00.067 PTS  2000  00:00:00.067 dur  1000  00:00:00.033 Offset 0x0c6c Size
Boxed Metadata 1 item
[0]  Key(4): blue Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 30
                    [ 4] DTS  3000  00:00:00.100 PTS  3000  00:00:00.100 dur  1000  00:00:00.033 Offset 0x0c84 Size
Boxed Metadata 3 items
[0]  Key(4): redd Keyspace(4): me4c  Value(16): 00 00 00 2C 00 00 00 00 00 00 00 14 00 00 00 30
[1]  Key(4): blue Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 00 00 00 00 14 00 00 00 30
[2]  Key(4): whte Keyspace(4): me4c  Value(16): 00 00 00 14 00 00 00 00 00 00 00 18 00 00 00 30
                    [ 5] DTS  4000  00:00:00.133 PTS  4000  00:00:00.133 dur  1000  00:00:00.033 Offset 0x0ccc Size
Boxed Metadata 3 items
[0]  Key(4): blck Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 10
[1]  Key(4): redd Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 10 00 00 00 40 00 00 00 10
[2]  Key(4): ylow Keyspace(4): me4c  Value(16): 00 00 00 00 00 00 00 20 00 00 00 40 00 00 00 10

...
```

Summary

- Successful file format
    - Very versatile: from editing to HTTP streaming to broadcasting
    - Very extensible (codecs, usages, applications)
    - Highly dynamic (the File Format group consistently receives numerous contributions at every MPEG meeting)
- Some challenges
    - Carrying some legacy that is no longer in use
    - Addressing all the use cases while maintain compatibility
    - For certain applications and use cases, the file format principles are suboptimal in terms of overhead or processing efficiency
- The ISO BMFF is a reliable foundation linking modern media and transport
- Continues to evolve within MPEG