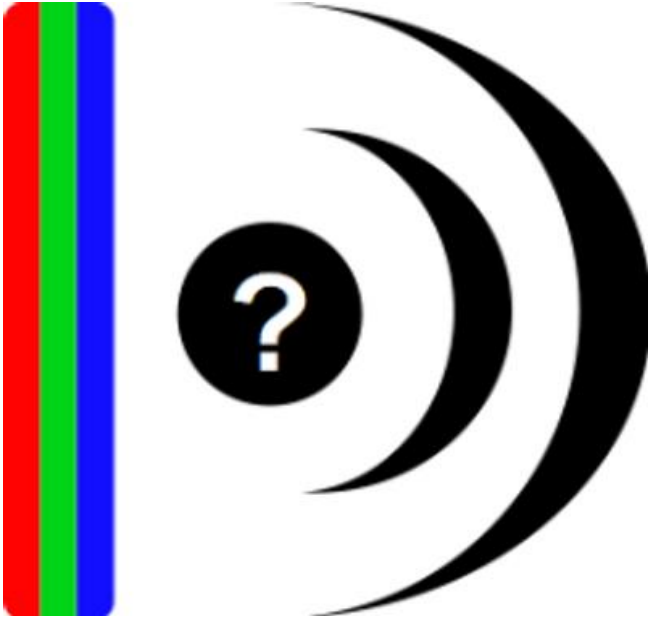# Checking Audiovisual Conformance

@JeromeM78 @dericed @MediaConch
#iasa_web 2017-09-20

MediaArea Team

Project Leaders:

Jérôme Martinez (Digital media analysis specialist)

Dave Rice (Archivist)

Other Members:

Guillaume Roques (Web developer)

Maxime Gervais (Junior developer / automation)

Ashley Blewer (Archivist)

Tessa Fallon (Archivist)

MediaArea contribution to open source

We want to spread open source! → Any platform and distribution kind.
We built an automated release system compatible with all tools we worked on.

Tools: MediaInfo, MediaConch, QCTools, BWF MetaEdit, AVI MetaEdit, MOV MetaEdit, DV Analyzer

for: Direct Download for Windows/Mac/Linux, RPM/DEB repository, Mac App Store (Mediainfo/MediaConch/QCTools only, others on the way), directly in Debian/Ubuntu/Fedora repos (MediaInfo/MediaConch only), HomeBrew, LinuxBrew

Interfaces: GUI, WebUI, command line, server mode, library (DLL/SO/DyLib)

PREFORMA Challenge

Empower memory institutions to gain full control over the technical properties of digital content intended for long-term preservation.

Open standard?	AUDIOVISUAL			TEXT	IMAGE
	broadcast	film	Consumer		
PREFORMA stakeholders	MPEG-IMX (MXF/MPEG2) XDCAM HD422 (MXF/MPEG4)	DPX DCP (MXF/JPEG2000)	MOV/MPEG2 AVI/MPEG2 MPEG/ MPEG2 MPEG/ MPEG4-AVC	PDF 1.4 PDF/A1	TIFF 6.0 JPEG JPEG2000 RAW
Industry standards	AS|07 (MXF/MPEG2) (MXF/JPEG2000) FIMS (MXF/MPEG2)	DCDM (TIFF 6.0) DCP (MXF/JPEG2000) IMF (MXF/MPEG4)	MPEG-AF	PDF	JPEG2000 TIFF
Open standards	MKV/FFV1 OGG/Dirac	PNG	WebM/VP8 OGG/Theora	PDF/A1 PDF/A3 PDF/A3	PNG
PREFORMA	MKV (?)	OGG	JPEG2000 (?)	FFV1	Dirac	LPCM (?)			PDF/A	TIFF 6.0 (?)

PREFORMA Projects

- MediaConch with focus on Matroska, FFV1, PCM

- VeraPDF with focus on PDF

- DPF Manager with focus on TIFF

MediaConch

Conformance Checker:

- Implementation Checker
- Policy Checker
- Reporter
- Fixer

Tour of MediaInfo and MediaTrace

MediaTrace sample output

```
<block offset="1920" name="Television information" size="128">
    <data offset="1920" name="SMPTE time code">4294967295</data>
    <data offset="1924" name="SMPTE user bits">4294967295</data>
    <data offset="1928" name="Interlace" moreinfo="2:1 interlace">255</data>
    <data offset="1929" name="Field number">255</data>
    <data offset="1930" name="Video signal standard" moreinfo="Undefined">0</data>
    <data offset="1931" name="Zero">255</data>
    <data offset="1932" name="Horizontal sampling rate (Hz)">0.000</data>
    <data offset="1936" name="Vertical sampling rate (Hz)">0.000</data>
    <data offset="1940" name="Temporal sampling rate or frame rate (Hz)">0.000</data>
    <data offset="1944" name="Time offset from sync to first pixel (ms)">0.000</data>
    <data offset="1948" name="Gamma">0.000</data>
    <data offset="1952" name="Black level code value">0.000</data>
    <data offset="1956" name="Black gain">0.000</data>
    <data offset="1960" name="Breakpoint">0.000</data>
    <data offset="1964" name="Reference white level code value">0.000</data>
    <data offset="1968" name="Integration time (s)">0.000</data>
```

EBML & Matroska

- Extensible Binary Meta Language (EBML is a Binary XML format)

- An EBML Schema defines an EBML Document like an XML Schema defines an XML Document

- Matroska and webm are EBML Document Type

- Storage is based on a structure of Element ID, Element Data Size, and Element Data

- Unlike XML, an EBML Document requires an EBML Schema to be interpreted semantically

```xml
<element name="FlagInterlaced" path="1*1(\Segment\Tracks\TrackEntry\Video\FlagInterlaced)"
cppname="VideoFlagInterlaced" id="0x9A" type="uinteger" minOccurs="1" maxOccurs="1" minver="2" webm="1" default="0"
range="0-2">
  <documentation lang="en">A flag to declare is the video is known to be progressive or interlaced and if applicable
to declare details about the interlacement.</documentation>
  <restriction>
    <enum value="0" label="undetermined"/>
    <enum value="1" label="interlaced"/>
    <enum value="2" label="progressive"/>
  </restriction>
</element>
<element name="FieldOrder" path="1*1(\Segment\Tracks\TrackEntry\Video\FieldOrder)" cppname="VideoFieldOrder"
id="0x9D" type="uinteger" minOccurs="1" maxOccurs="1" minver="4" webm="0" default="2" range="0-14">
  <documentation lang="en">Declare the field ordering of the video. If FlagInterlaced is not set to 1, this Element
MUST be ignored.</documentation>
  <restriction>
    <enum value="0" label="progressive"/>
    <enum value="1" label="tff">
      <documentation lang="en" type="definition">Top field displayed first. Top field stored first.</documentation>
    </enum>
    <enum value="2" label="undetermined"/>
    <enum value="6" label="bff">
      <documentation lang="en" type="definition">Bottom field displayed first. Bottom field stored
first.</documentation>
    </enum>
    <enum value="9" label="bff(swapped)">
      <documentation lang="en" type="definition">Top field displayed first. Fields are interleaved in storage with
the top line of the top field stored first.</documentation>
    </enum>
    <enum value="14" label="tff(swapped)">
      <documentation lang="en" type="definition">Bottom field displayed first. Fields are interleaved in storage with
the top line of the top field stored first.</documentation>
    </enum>
  </restriction>
</element>
```

Element Name	L	EBML ID	Ma	Mu	Rng	Default	T	1	2	3	4	W	Description
EBML Header													
EBML	0	[1A][45][DF][A3]	mand.	mult.	-	-	m	*	*	*	*	*	Set the EBML characteristics of the data to follow. Each EBML document has to start with this.
EBMLVersion	1	[42][86]	mand.	-	-	1	u	*	*	*	*	*	The version of EBML parser used to create the file.
EBMLReadVersion	1	[42][F7]	mand.	-	-	1	u	*	*	*	*	*	The minimum EBML version a parser has to support to read this file.
EBMLMaxIDLength	1	[42][F2]	mand.	-	-	4	u	*	*	*	*	*	The maximum length of the IDs you'll find in this file (4 or less in Matroska).
EBMLMaxSizeLength	1	[42][F3]	mand.	-	-	8	u	*	*	*	*	*	The maximum length of the sizes you'll find in this file (8 or less in Matroska). This does not override the element size indicated at the beginning of an element. Elements that have an indicated size which is larger than what is allowed by EBMLMaxSizeLength shall be considered invalid.
DocType	1	[42][82]	mand.	-	-	matroska	s	*	*	*	*	*	A string that describes the type of document that follows this EBML header. 'matroska' in our case or 'webm' for webm files.
DocTypeVersion	1	[42][87]	mand.	-	-	1	u	*	*	*	*	*	The version of DocType interpreter used to create the file.
DocTypeReadVersion	1	[42][85]	mand.	-	-	1	u	*	*	*	*	*	The minimum DocType version an interpreter has to support to read this file.

EBML & Matroska

```
<EBML>
        <EBMLVersion>1</EBMLVersion>
        <EBMLReadVersion>1</EBMLReadVersion>
        <EBMLMaxIDLength>4</EBMLMaxIDLength>
        <EBMLMaxSizeLength>8</EBMLMaxSizeLength>
        <DocType>matroska</DocType>
        <DocTypeVersion>4</DocTypeVersion>
        <DocTypeReadVersion>2</DocTypeReadVersion>
</EBML>
<Segment>
        <Info>
        <CRC-32>54272a11</CRC-32>
                <TimecodeScale>1000000</TimecodeScale>
                <MuxingApp>Lavf57.50.100</MuxingApp>
                <WritingApp>Lavf57.50.100</WritingApp>
                <SegmentUID>efd46d3ed630381ef9021d1d4ed5a81a</SegmentUID>
                <Duration>40.0</Duration>
        </Info>
        <Tracks>
```

FFV1 perks

- Losslessness
- Fixity
- Self-description
- Size

“[ffv1 @ 0x7f9855046e00] CRC mismatch FC686A4F! frame 215”

/Users/drice/Desktop/ffv1_tests/DBS00018−12.ffv1.3.g1.v0.fps71.fuzz.mov

```xml
<block offset="499" name="Private data" size="41">
 <block offset="499" name="ConfigurationRecord" size="41">
  <block offset="499" name="Parameters" size="0">
   <data offset="500" name="version">3</data>
   <data offset="501" name="micro_version">4</data>
  </block>
  <data offset="501" name="coder_type">0</data>
  <data offset="501" name="colorspace_type">0</data>
  <data offset="501" name="bits_per_raw_sample">0</data>
  <data offset="501" name="chroma_planes">Yes</data>
  <data offset="502" name="log2(h_chroma_subsample)">1</data>
  <data offset="502" name="log2(h_chroma_subsample)">0</data>
  <data offset="502" name="alpha_plane">No</data>
  <data offset="502" name="num_h_slices_minus1">1</data>
  <data offset="502" name="num_v_slices_minus1">1</data>
  <data offset="503" name="quant_table_count">2</data>
  <block offset="499" name="QuantizationTableSet" size="0">
   <block offset="499" name="QuantizationTable" size="0">
    <data offset="503" name="len_minus1">0</data>
    <data offset="503" name="len_minus1">0</data>
    <data offset="504" name="len_minus1">2</data>
    <data offset="504" name="len_minus1">6</data>
    <data offset="506" name="len_minus1">22</data>
    <data offset="507" name="len_minus1">92</data>
   </block>
   <block offset="499" name="QuantizationTable" size="0">
```

```xml
<block offset="1196" name="Data" parser="FFV1" size="138575">
  <block offset="1196" name="Frame" size="138575">
    <data offset="1197" name="keyframe">Yes</data>
    <block offset="1196" name="Slice" size="31291">
      <block offset="1196" name="SliceHeader" size="0">
        <data offset="1197" name="slice_x">0</data>
        <data offset="1197" name="slice_y">0</data>
        <data offset="1197" name="slice_width_minus1">0</data>
        <data offset="1197" name="slice_height_minus1">0</data>
        <data offset="1197" name="quant_table_index">0</data>
        <data offset="1197" name="quant_table_index">0</data>
        <data offset="1198" name="picture_structure">1</data>
        <data offset="1199" name="sar_num">9</data>
        <data offset="1200" name="sar_den">10</data>
        <data offset="1200" name="end">0</data>
      </block>
      <block offset="1196" name="SliceContent" size="31283"/>
      <block offset="32479" name="SliceFooter" size="8">
        <data offset="32479" name="slice_size">31283</data>
        <data offset="32482" name="error_status">0</data>
        <data offset="32483" name="slice_crc_parity">2388880507</data>
      </block>
    </block>
    <block offset="32487" name="Slice" size="31188">
      <block offset="32487" name="SliceHeader" size="0">
        <data offset="32488" name="slice_x">1</data>
        <data offset="32488" name="slice_y">0</data>
        <data offset="32488" name="slice_width_minus1">0</data>
        <data offset="32488" name="slice_height_minus1">0</data>
        <data offset="32488" name="quant_table_index">0</data>
        <data offset="32488" name="quant_table_index">0</data>
        <data offset="32489" name="picture_structure">1</data>
        <data offset="32490" name="sar_num">9</data>
        <data offset="32491" name="sar_den">10</data>
        <data offset="32491" name="end">0</data>
      </block>
      <block offset="32487" name="SliceContent" size="31180"/>
      <block offset="63667" name="SliceFooter" size="8">
        <data offset="63667" name="slice_size">31180</data>
        <data offset="63670" name="error_status">0</data>
        <data offset="63671" name="slice_crc_parity">2749361047</data>
      </block>
    </block>
  </block>
```

Tour of MediaConch

Implementation

Table 4–2 Video sample description extensions

Extension type	Description
`'gama'`	A 32–bit fixed–point number indicating the gamma level at which the image was captured. The decompressor can use this value to gamma–correct at display time.
`'fiel'`	Two 8–bit integers that define field handling. This information is used by applications to modify decompressed image data or by decompressor components to determine field display order. This extension is mandatory for all uncompressed Y´CbCr data formats. The first byte specifies the field count, and may be set to 1 or 2. A value of 1 is used for progressive–scan images; a value of 2 indicates interlaced images. When the field count is 2, the second byte specifies the field ordering: which field contains the topmost scan–line, which field should be displayed earliest, and which is stored first in each sample. Each sample consists of two distinct compressed images, each coding one field: the field with the topmost scan–line, T, and the other field, B. The following defines the permitted variants: 0 – There is only one field. 1 – T is displayed earliest, T is stored first in the file. 6 – B is displayed earliest, B is stored first in the file. 9 – B is displayed earliest, T is stored first in the file. 14 – T is displayed earliest, B is stored first in the file.
`'mjqt'`	The default quantization table for a Motion–JPEG data stream.

dwsinger commented on Aug 12

Ah, not quite. 1 and 6 are indeed 'planar' (all of one field before all of the other). They don't concern us. Both 9 and 14 are stored in spatial order (i.e. you could do terrible de-interlacing by simply displaying the buffer as a frame), and the 9 or 14 value tells you which field is to be displayed first.

9 – T is earlier than B. 14 – B is earlier than T

This is, afaik, what we do; the file format documentation appears to be, um, guilty of, well, wrongness. If you google "letters from the ice-floe" for Dispatch 19, this is all spelled out in the goriest of detail.

😄 2

Kieran Kunhya
@kierank_

I made a table to show the difference between Cineform specs and reality (cc @FFmpeg, @libav_org and @videolan)

Value of ImageFormat Tag (tag=84)

	VC-5 Spec	Reality (libavcodec)
YUV 4:2:2	2	1
RGB 12-bit	N/A	3
RGBA 12-bit	1	4
Bayer	3	2

11:17 AM - 28 Feb 2016

6 Retweets **5** Likes

3   6   5

Policy Example: Apple's TN2162 (Uncompressed MOV)

Using these labels, application and device developers can finally relieve end users of the following nagging usability problems:

- "I captured this file on a Mac but it looks dark on a PC."
- "I rendered to this file in app A; it plays ok in MoviePlayer but looks like snow in app B."
- "I captured this data and the colors look wrong on my computer screen."
- "I captured this data but the fields are swapped when I play it back. My application has 6 different field-related controls and I've tried all 64 combinations but none of them work."
- "I captured and played this data OK, but whenever I try to render or do effects using this data, there are stuttery forward-backward motion problems."
- "I captured this data and it looks squished or stretched horizontally."
- "Circles I lay down in my application don't look circular on the video monitor."
- "I am compositing similar video footage which I had captured with two different devices, and the video data is shifted horizontally. No setting of captured image size seems to help."

Tour of MediaConch

Policy
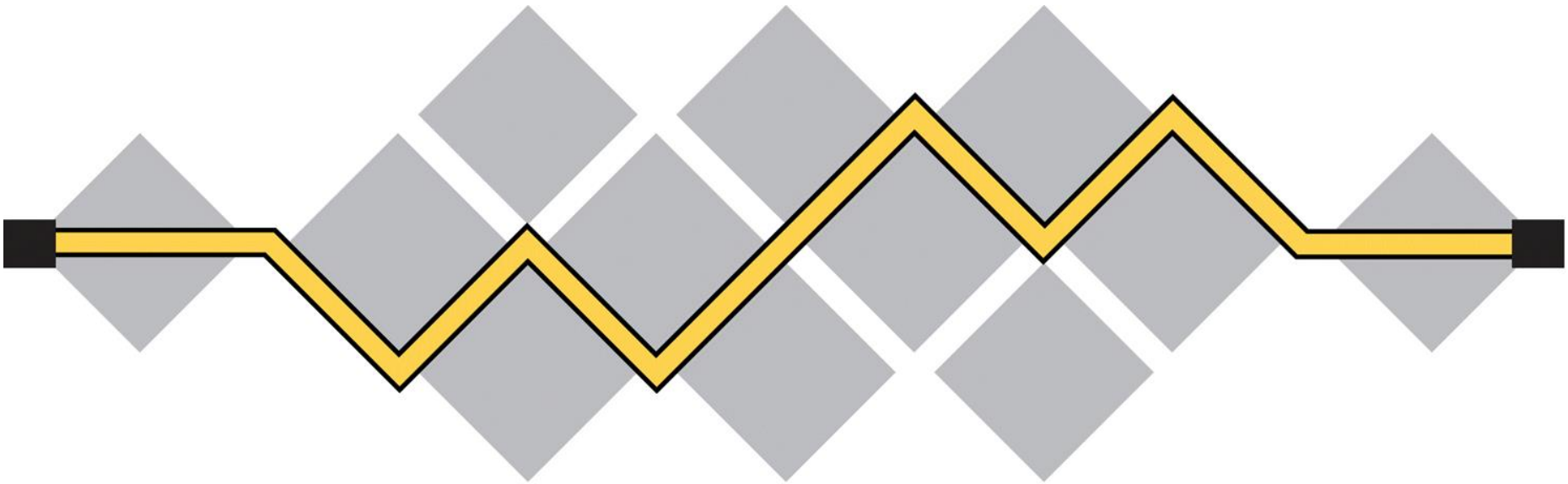
CELLAR
Codec Encoding for LossLess Archiving and Realtime transmission

Project Advisors

MATROŠKA {}

FFmpeg

libav

artefactual

Codec Encoding for LossLess Archiving and Realtime transmission (cellar)

Documents | **Charter** | Meetings | History | Photos | Email expansions | List archive | Tools »

WG	**Name**	Codec Encoding for LossLess Archiving and Realtime transmission
	Acronym	cellar
	Area	Applications and Real-Time Area (art)
	State	Active
	Charter	charter-ietf-cellar-01 `Approved`
	Dependencies	Document dependency graph (SVG)
Personnel	**Chairs**	✉ Tessa Fallon
		✉ Tim Terriberry
	Area Director	✉ Ben Campbell
Mailing list	**Address**	cellar@ietf.org
	To subscribe	https://www.ietf.org/mailman/listinfo/cellar
	Archive	https://mailarchive.ietf.org/arch/browse/cellar/
Jabber chat	**Room address**	xmpp:cellar@jabber.ietf.org?join
	Logs	https://jabber.ietf.org/logs/cellar/

Charter for Working Group

The preservation of audiovisual materials faces challenges from technological obsolescence, analog media deterioration, and the use of proprietary formats that lack formal open standards. While obsolescence and material degradation are widely addressed, the standardization of open, transparent, self-descriptive, lossless formats remains an important mission to be undertaken by the open source community.

FFV1 is a lossless video codec and Matroska is an extensible media container based on EBML (Extensible Binary Meta Language), a binary XML format. There are open source implementations of both formats, and an increasing interest in and support for use of FFV1 and Matroska. However, there are concerns about the sustainability and credibility of existing specifications for the long-term use of these formats. These existing specifications require broader review and formalization in order to encourage widespread adoption.

There is also a need for a lossless audio format to complement the lossless video codec and container format. FLAC is a lossless audio codec that has seen widespread adoption in a number of different applications including archival applications. While there are open source implementations of the codec, no formal standards for either the codec itself or its use in container formats currently exist. Review and formalization of the FLAC codec standard and its use in Matroska container formats is needed for wider adoption.

Using existing work done by the development communities of Matroska, FFV1, and FLAC, the Working Group will formalize specifications for these open and lossless formats. In order to provide authoritative, standardized specifications for users and developers, the Working Group will seek consensus throughout the process of refining and formalizing these standards. Initial specifications can be accessed here:

Codec Encoding for LossLess Archiving and Realtime transmission (cellar)

Document	Date	Status	IPR	AD / Shepherd
Active Internet-Drafts				
draft-ietf-cellar-ebml-03 **Extensible Binary Meta Language**	2017-07-03 38 pages	I-D Exists WG Document: Informational		
draft-ietf-cellar-ffv1-00 **FF Video Codec 1**	2017-07-03 39 pages	I-D Exists Waiting for WG Chair Go-Ahead		
draft-niedermayer-cellar-ffv1-02 **FF Video Codec 1**	2017-05-09 36 pages	I-D Exists WG Document: Informational *Apr 2017*		

Document	Date	Status	IPR	AD / Shepherd
Related Internet-Drafts				
draft-lhomme-cellar-matroska-03 **Matroska**	2017-07-03 216 pages	I-D Exists *Sep 2017*		
draft-xiph-cellar-flac-00 **Free Lossless Audio Codec**	2017-06-05 31 pages	I-D Exists *Dec 2017*		

What is Archivematica?

An open-source platform for digital preservation which packages digital objects for long-term storage. Archivematica makes AIPs (Archival Information Packages).
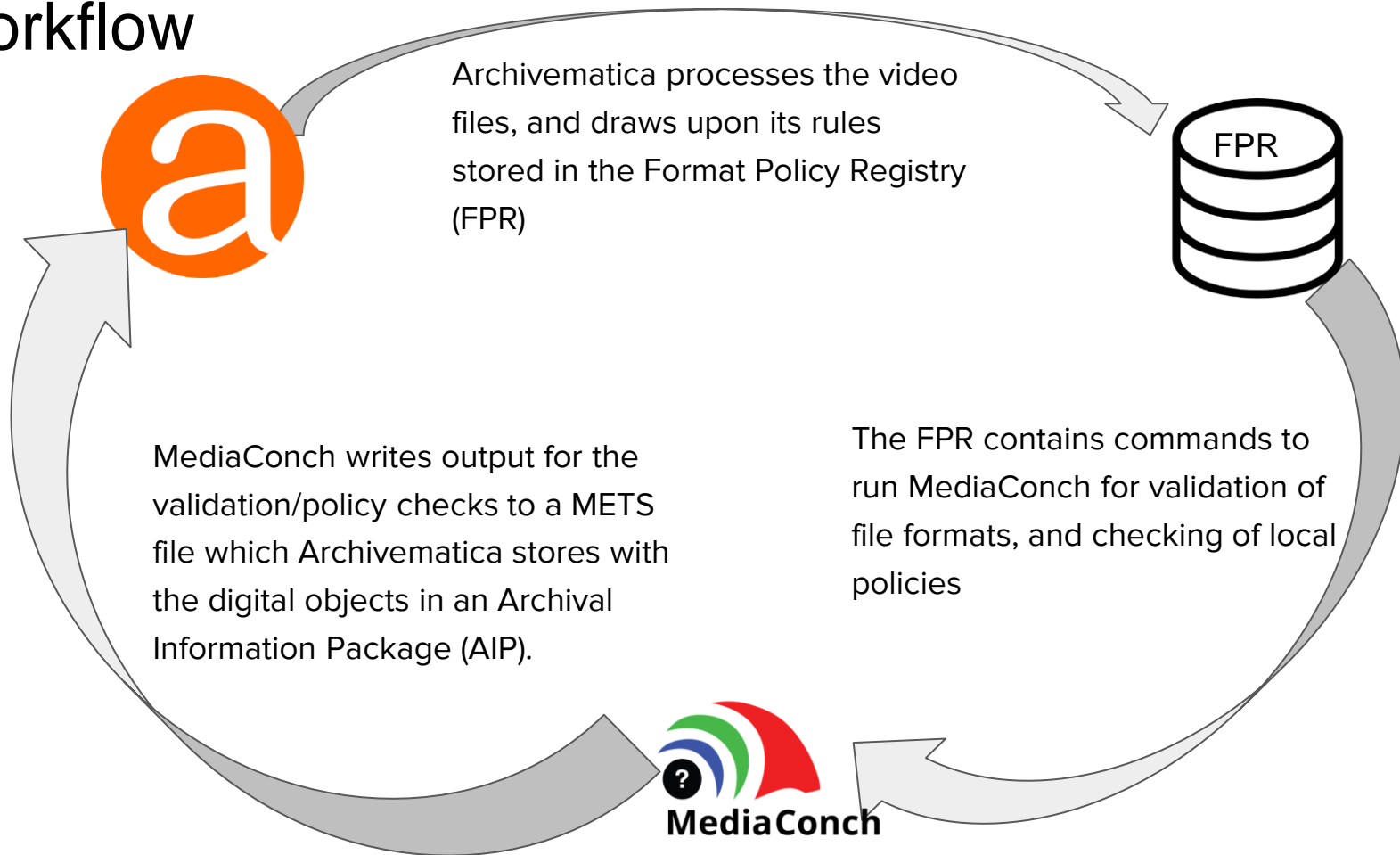
What is MediaConch?

An open-source implementation and policy checker for Matroska* and FFV1** files. Developed by MediaArea (see: MediaInfo) and funded by PREFORMA.

Achieves two things: 1., are my files valid according to the specification? And 2., do they conform to our local policies?

*A container format for video, audio, related streams
**A lossless video encoding

Workflow

Archivematica processes the video files, and draws upon its rules stored in the Format Policy Registry (FPR)

FPR

The FPR contains commands to run MediaConch for validation of file formats, and checking of local policies

MediaConch writes output for the validation/policy checks to a METS file which Archivematica stores with the digital objects in an Archival Information Package (AIP).

MediaConch

Update one: made new commands

Update two: made new rules

Update six: METS-all-the-things

```xml
    </mets:mdWrap>
  </mets:digiprovMD>
 -<mets:digiprovMD ID="digiprovMD_4">
  -<mets:mdWrap MDTYPE="PREMIS:EVENT">
   -<mets:xmlData>
    -<premis:event xsi:schemaLocation="info:lc/xmlns/premis-v2 http://www.loc.gov/standards/premis/v2/premis-v2-2.xsd" version="2.2">
      -<premis:eventIdentifier>
         <premis:eventIdentifierType>UUID</premis:eventIdentifierType>
         <premis:eventIdentifierValue>32e14f48-b90b-4c24-9f70-e6b2660445b6</premis:eventIdentifierValue>
       </premis:eventIdentifier>
         <premis:eventType>validation</premis:eventType>
         <premis:eventDateTime>2016-10-05T21:20:31+00:00</premis:eventDateTime>
         <premis:eventDetail>program="MediaConch"; version="16.05"</premis:eventDetail>
      -<premis:eventOutcomeInformation>
         <premis:eventOutcome>pass</premis:eventOutcome>
        -<premis:eventOutcomeDetail>
         -<premis:eventOutcomeDetailNote>
            MediaConch policy check result: All policy checks passed: Video BitDepth equals 8 (bits); Audio BitDepth is greater or equal than 16-bit;
            Audio Channels are greater or equal than 1; Video ColorSpace equals YUV; Audio Format equals PCM; Video CodecID equals FFV1; Video
            Width equals 720 (pixels); Video Height equals 486 (pixels); Video DisplayAspectRatio equals 4:3 (1.333); Video ChromaSubsampling equals
            4:2:0; General Format equals Matroska; Audio SamplingRate is greater or equal than 48 kHz (48000); Video Format is FFV1
           </premis:eventOutcomeDetailNote>
          </premis:eventOutcomeDetail>
         </premis:eventOutcomeInformation>
       -<premis:linkingAgentIdentifier>
          <premis:linkingAgentIdentifierType>preservation system</premis:linkingAgentIdentifierType>
          <premis:linkingAgentIdentifierValue>Archivematica-1.5</premis:linkingAgentIdentifierValue>
         </premis:linkingAgentIdentifier>
       -<premis:linkingAgentIdentifier>
          <premis:linkingAgentIdentifierType>repository code</premis:linkingAgentIdentifierType>
          <premis:linkingAgentIdentifierValue>artefactual</premis:linkingAgentIdentifierValue>
```

FFV1 in VIAA

Test on real files
Speed tests (FFmpeg FFV1 decoding faster than FFmpeg JP2k decoding)
Automation of transcoding from MXF/JP2k/PCM to MKV/FFV1/FLAC + derivate files, then conformance checking and test of framehash of MKV/FFV1/FLAC files.
Compression ratio tests

Report: https://viaa.be/files/attachments/.669/VIAA_Preservation_Reformatting.pdf

MediaInfo

"MIXML": working on a true XML output, less tied to text output

EBUCore 1.8, including Sony Acquisition Metadata full parsing

EBUCore 1.8 JSON output

HDR Metadata (preferred_transfer_characteristics, MaxCLL, MaxFALL, MasteringDisplay_ColorPrimaries, MasteringDisplay_Luminance)

Example of 608/708 extraction and decoding from MXF/ANC/CDP

BWF MetaEdit

Supports embedding, validating, and exporting of metadata in Broadcast WAVE Format (BWF) files.
It supports the FADGI (Federal Agencies Digitization Guidelines Initiative) for embedded metadata in the bext and INFO chunks.

Initially funded by the Library of Congress and FADGI
designed and led by AVPreserve
developed by MediaArea.

BWF MetaEdit update by MediaArea (soon!)

General maintenance:

Update of third party libs (Qt)

Drag and drop was not working on newest versions of macOS

Support of latest versions of Debian, Ubuntu, Fedora, RHEL/CentOS...

Bug fixes:

Fix sync issues between samples count and translated timestamp

Unknown loudness was stored as 0x0000 but specs says it must be 0x7FFF

CodingHistory edition was not working well on macOS

CodingHistory was not correctly reading history with carriage returns

Fix crash with some files

Improvements:

Commmand line option for outputing XML report to a specific file

MOV MetaEdit

New project

Code on GitHub, snapshots available

Release in October 2017

Currently focused on 2 items:

- add/edition of the "pasp" atom (anonymously sponsored, CLI only)

- add/edition/delete of "Universal Ad ID" metadata (sponsored by Ad-ID)
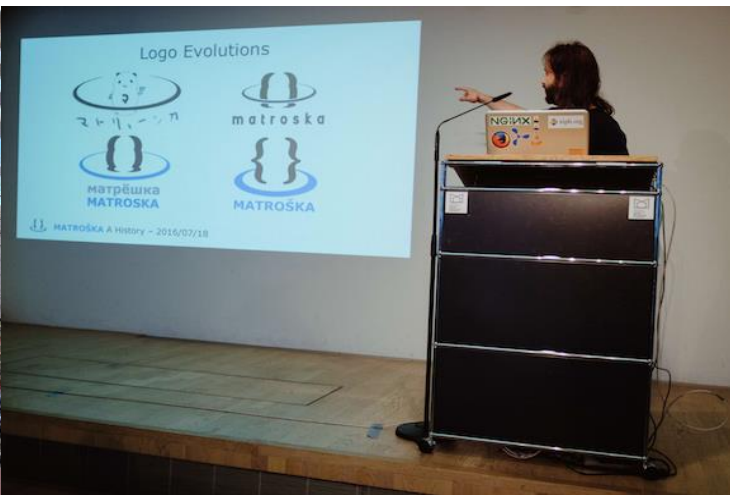
Core of the program built for extensibility, can be extended depending of sponsor

Specification Development via GitHub

- https://github.com/Matroska-Org/ebml-specification

- https://github.com/Matroska-Org/matroska-specification

- https://github.com/FFmpeg/FFV1/

- https://github.com/xiph/flac

- https://www.ietf.org/mailman/listinfo/cellar

No Time To Wait 2016 - An Matroska & FFV1 Symposium

No Time To Wait 2017 - November 9/10 Vienna
"open media, standardization, and audiovisual preservation"
https://mediaarea.net/MediaConch/notimetowait2.html

Stay in touch

MediaArea: https://mediaarea.net, @MediaArea_net
MediaConch: https://mediaarea.net/MediaConch, @MediaConch

Jérôme Martinez: jerome@mediaarea.net
Dave Rice: https://twitter.com/dericed

Slides: https://mediaarea.net/Events

Archivematica related slides from Artefactual