MPEG File Formats

MPEG-4 Part 12, 14, 15
MPEG-21 Part 9

Cyril Concolato (ENST)
cyril.concolato@enst.fr

March 14th, 2005

Background

- Development of MPEG-4 => storage needs
  - MPEG-4 = possibly many elementary streams
  - Different from MPEG-2
- MPEG Call for Proposal
  - Chosen solution : Apple et Quicktime file format (.mov)
  - Basis for the MP4 file format and derivatives
- Call for Proposal for Motion JPEG-2000
  - Chosen solution : Apple et Quicktime file format (.mov)
  - Basis for MJP2 format
- Decision to create a common specification
  - The ISO Base Media File Format
- Call for Proposal for H264 FF and MPEG-21 FF
  - Results: extensions of the ISO Base Media FF

Organization of the specifications

- MPEG-4 Part 12
  - Base specification common to MP4, 3GP, MJP2, MP21, AVC
- MPEG-4 Part 14
  - Specific extensions for MPEG-4 content
- MPEG-4 Part 15
  - Specific extensions for AVC
- MPEG-21 Part 9
  - Extensions and rules to embed MPEG-21 DID and resources in an MP21 file.

MPEG File Format Principles (1/3)

- Binary, compact format
- Hierarchical and structured format
- Extensible: backward- & forward-compatibility
- Designed initially to store timed content
- Designed to ease content exchange
  - A file can be self-contained
    - The file contains all the data to be exchanged
  - Small Size
    - The file only contains the data to be exchanged.

MPEG File Format Principles (2/3)

- Efficient Editing
  - By storing each stream separately (in a track)
  - By possibly editing several streams together
  - By minimizing the disc accesses
- Easy streaming
  - Ability to store streaming directives, for any protocol, together with the data to be streamed
  - Ability to store transport packets prepared in advance
- Efficient playback of local files
  - Ability to interlace streams to minimize the disc accesses
  - Allow fast search in the file for fast rewind/forward operations
- Ability to download and play the file at the same time

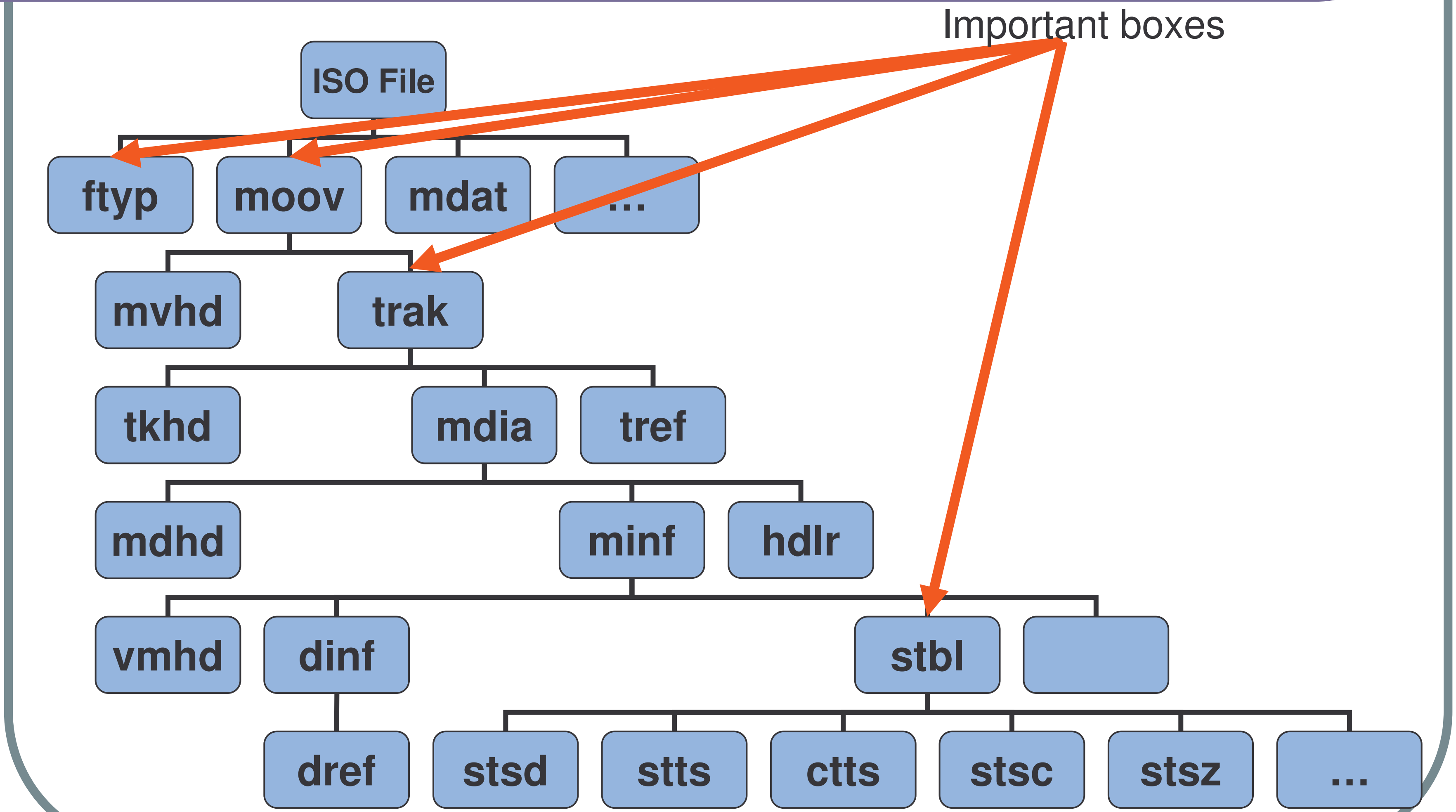
MPEG File Format Principles (3/3)

- Object-oriented files
  - The structure of the object is determined by its type
  - Base structure, building-block = Box (Atom in early specs)
    - Type identifier on 4 bytes
    - Length of the box on 4 bytes
    - Data specific to the box type, including other boxes on N bytes
  - No data is stored outside boxes
- Backward- & forward-compatible formats
  - A file reader conformant to an old version of the spec is able to read files conformant to a new version.
    - It detects unknown boxes (type)
    - It skips unknown boxes (length)
  - A new file reader can of course read old files
- Ability to spread the content over several files

Structure of a file

- Media-data and Meta-data are separate
  - Media-Data are Access Units or Samples (e.g. vidéo, audio, …)
  - Meta-data indicates:
    - Type of stream using a 4 bytes code
    - Timestamps (CTS, DTS) of the samples
    - Position & length of the samples in the file or in external files
    - Flag for random access (e.g. I frames) …
- Advantages
  - Non-linear Editing operations are eased
    - Editing of pointers vs editing of data
  - Fast search is as easy in big or small file
- Restrictions
  - If the media data are spread, one file needs to contain all the metadata

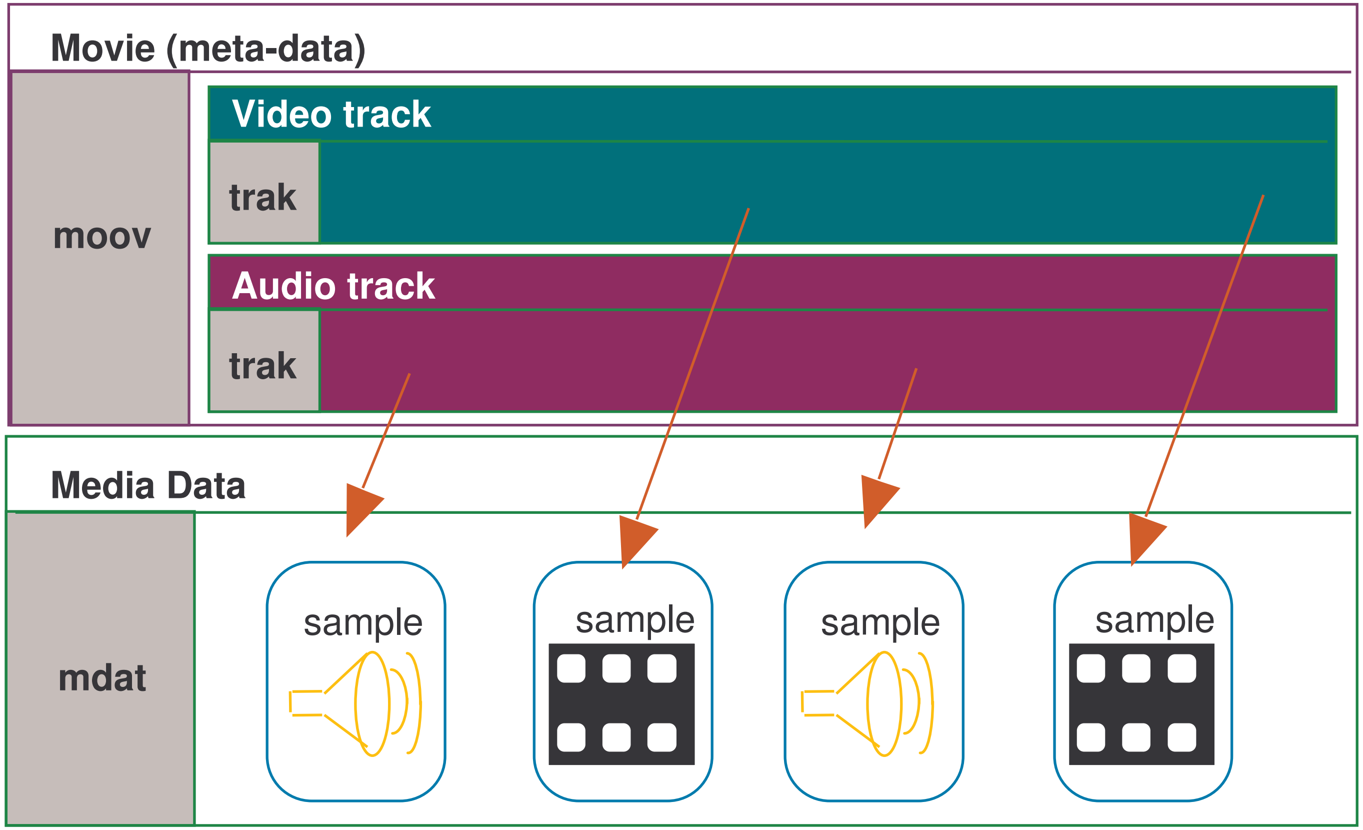
Typical Box Hierarchy



Some boxes …

- `ftyp` (File Type): 1 per file
  - File type
  - File version
  - Compatibility with other ISO files
- `mdat` (Media Data):
  - Contains the media data
  - A file may have several, non contiguous
- `moov` (Movie):
  - Unique container for the metadata of a presentation
- `mvhd` (Movie Header):
  - Generic info about the movie
- `trak` (Track):
  - Container for Meta-date related to one stream
- `hdlr` (Handler)
  - Indicates the type of stream
- `dinf/dref` (Data Information/Data Reference)
  - Indicates the location of the data (current file or remote file)
- `stbl` (Sample Table)
  - Contains the meta data related to samples, sample per sample
- `stsd` (Sample Description)
  - Contains the decoder configuration for the elementary stream
- `stts` (Sample To Time)
  - DTS for each sample
  - Use a predictive coding scheme
- `stsz` (Sample To Size)
  - Size of each sample, run-length coded

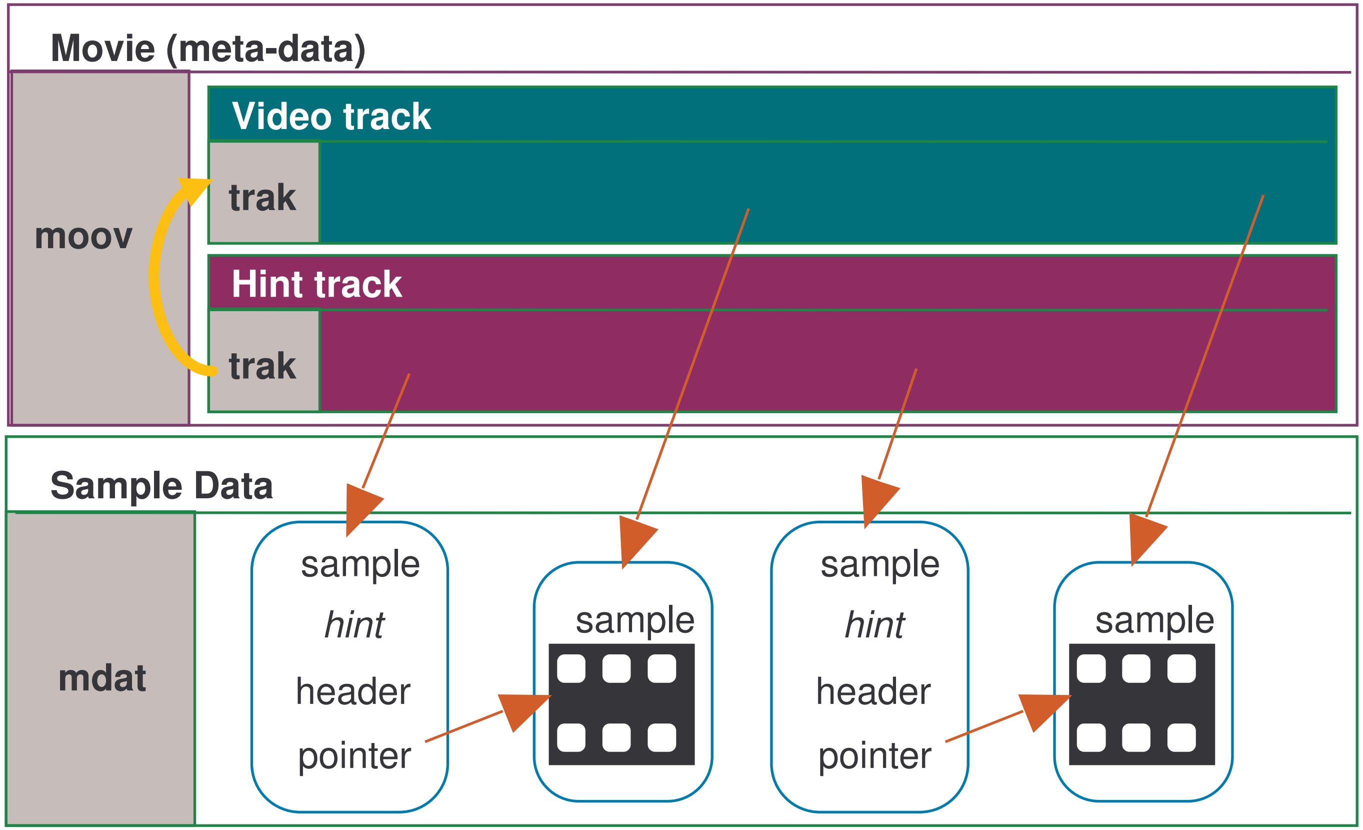
Example : an mp4 file



Usage scenarios

- Progressive download and play
  - `moov` box written before the `mdat` boxes
    - Reception of all the meta-data needed to play the data before reception of the data themselves
  - Interleaving of elementary streams in `mdat`
    - Enables reading the movie as the data arrives
- Streaming
  - Use of a special kind of track: the Hint track
  - Track linked to a media track
  - Gives the streaming directives to stream the referenced track
  - Design generic for all protocols
    - But a particular hint track is specific to one protocol (e.g. RTP)

Streaming & hint track



Usage scenarios

- On the fly creation of timed files
  - E.g. Capture of Live Audio
  - On low-end terminals, small memory
  - Need to write `moov` by chunk
  - => Use of Movie Fragments
    - `moov` followed by `moof` or `mfra` boxes
    - Use of `trun` instead of `trak`

MPEG-4 specifics

- MP4 extensions
  - Storage of `InitialObjectDescriptor`
    - Use of a high-level box: `iods`
  - Signalling of ElementaryStreamDescriptor
    - Use of track-level box `stsd`
  - Definition of new code points for MPEG-4 streams
    - BIFS, OD, ClockReference, MPEG-J, IPMP
- H264 extensions
  - Independent and disposable sample box
  - Notion of Sample groups
  - Parameter Set elementary stream structures

MPEG-21 Extensions

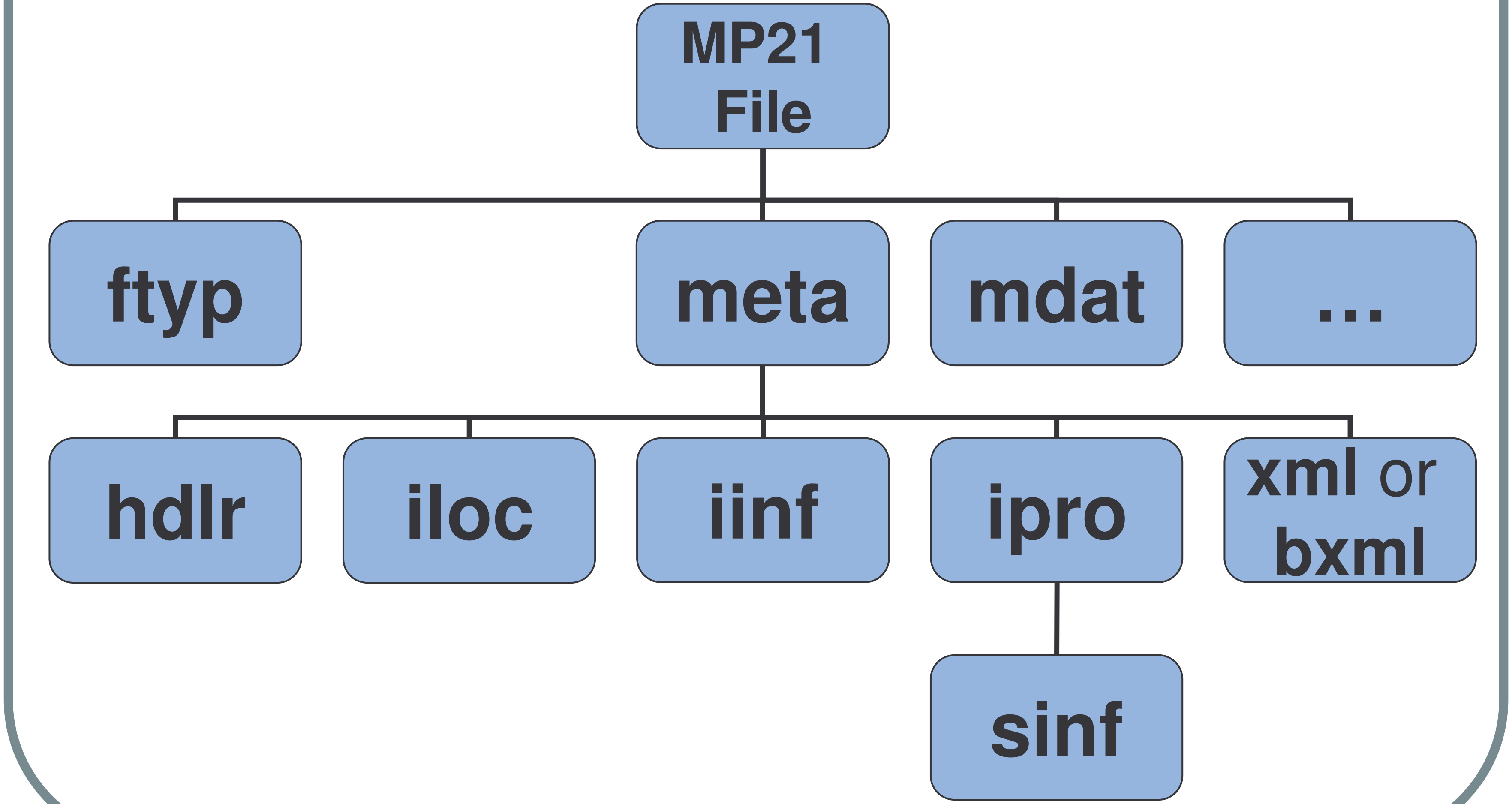
- General Need
  - to store descriptive or annotative metadata like XML data
  - E.g. DID, SMIL …
- MPEG-21 Need
  - to package DID + resources

⇒ Creation of the MPEG-21 file format

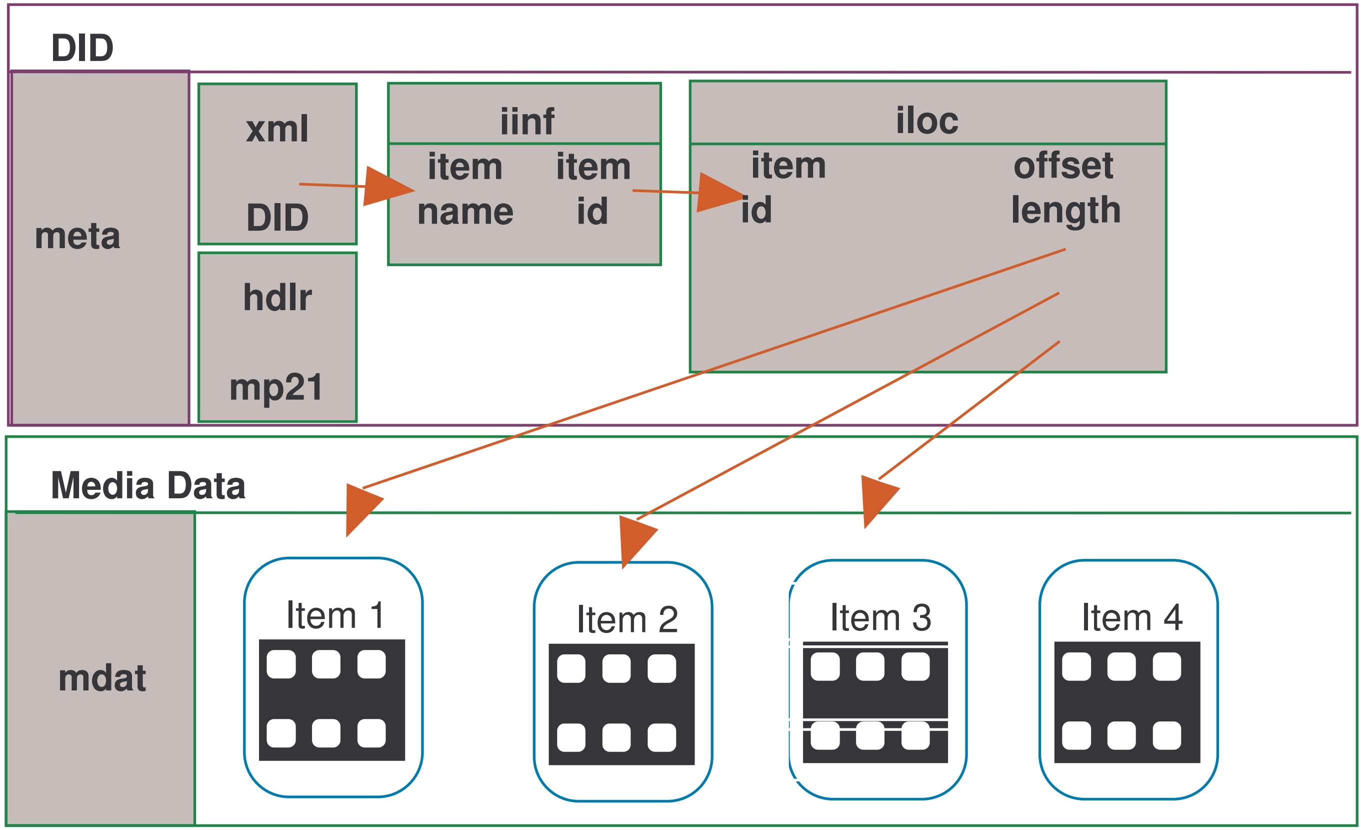
Specification of one high-level box: `meta`

  - To store "metadata" identified by the `hldr` box (4 bytes code)
    - Possibly XML or Binary XML data (`xml`, `bxml` boxes)
  - To inform about the resources embedded in the file (`iinf` box)
  - To enable efficient storage of the resources
    - Inside the file or outside of the file (reuse of `dinf/dref` boxes)
    - Using a directory of resources (`iloc` box)
  - To inform about the protection mechanism used (`ipro` box)
- Special case of MPEG-21 files: "dual-headed" files
  - DID describing an MPEG-4 resource
  - Readable by MPEG-4 player and MPEG-21 peer

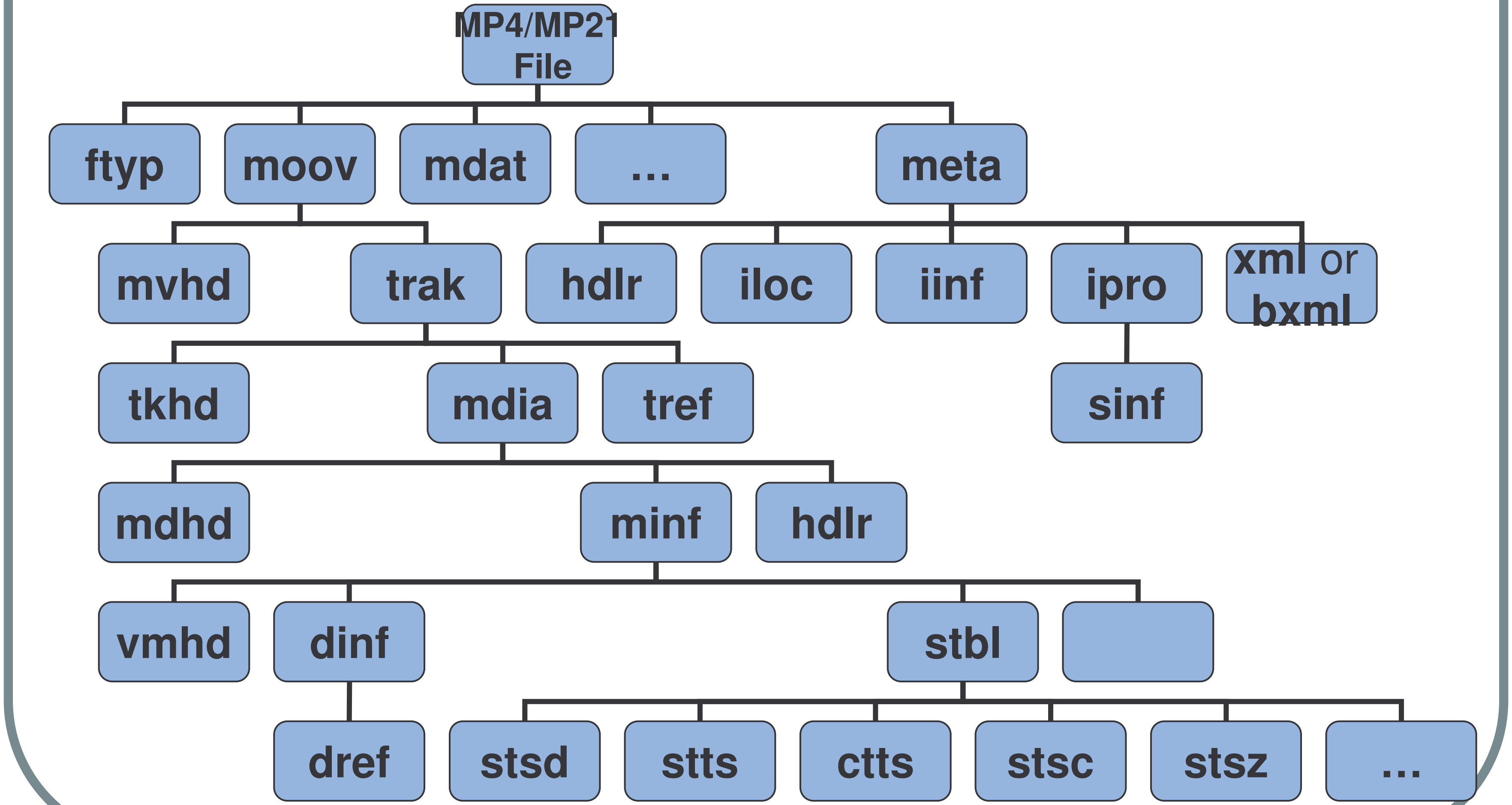
Typical hierachy of an MP21 file



Example : an mp21 file



Dual-headed file



Software

- ISO reference software
- Open Source tool, GPL license
  - MP4Box, MPEG-4 file reader/creator
    - http://gpac.sourceforge.net
  - MP21Box, MPEG-21 file reader/creator
    - http://perso.enst.fr/~concolat/MPEG-21/mp21box.htm