

# Containerformate am Beispiel MP4

Medientechnologie IL

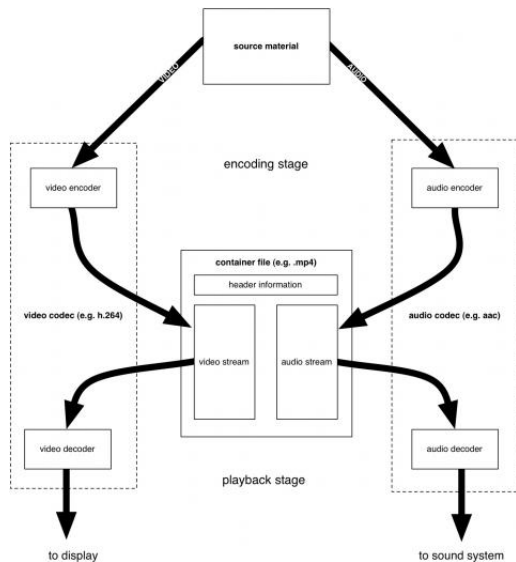
Andreas Unterweger

Vertiefung Medieninformatik  
Studiengang ITS  
FH Salzburg

Sommersemester 2017

- Container: Speicherung zusammengehöriger Audio-, Video- und anderer Ströme in einer gemeinsamen Struktur (z.B. Datei)
- Aufgaben
  - Strukturierte Speicherung
  - Synchronisation (der einzelnen Ströme)
  - Ermöglichung bzw. Vereinfachung von Random Access
  - Meta- und/oder Benutzerdatenspeicherung (optional)
- Containerformate spezifizieren Containeraufbau
- Formatspezifische Beschränkungen (Auswahl)
  - Unterstützte Audio-, Video- und andere Formate
  - Unterstützung von B-Frames (allgemein: Diskrepanz zwischen Kodier- und Darstellungsreihenfolge)
  - Unterstützung mehrerer Videostreams (z.B. für stereoskopisches 3-D)
  - Einzelbildgrößen-, Frameraten- und andere Beschränkungen
  - Unterstützung von Meta- und Benutzerdatenspeicherung

# Überblick Containerformate II



Quelle: <http://serato.com/video-s1/support/1931/video-files-101>

- Kodierung (englisch *Encoding*): Daten in bestimmtes Format bringen
- Dekodierung (englisch *Decoding*): Rohdaten aus kodierten Daten zurückgewinnen
- Transkodierung (englisch *Transcoding*): Daten von einem Format in ein anderes bringen (Dekodierung plus Kodierung)
- Multiplexing (kurz Muxing): Kodierte Daten (z.B. Audio und Video) in Container (mit bestimmtem Format) zusammenfassen
- Demultiplexing (kurz Demuxing): Kodierte Daten (z.B. Audio und Video) aus zusammengefassten rückgewinnen
- Transmuxing: Zusammengefasste Daten in anderes zusammengefasstes Format bringen (Demuxing plus Muxing)

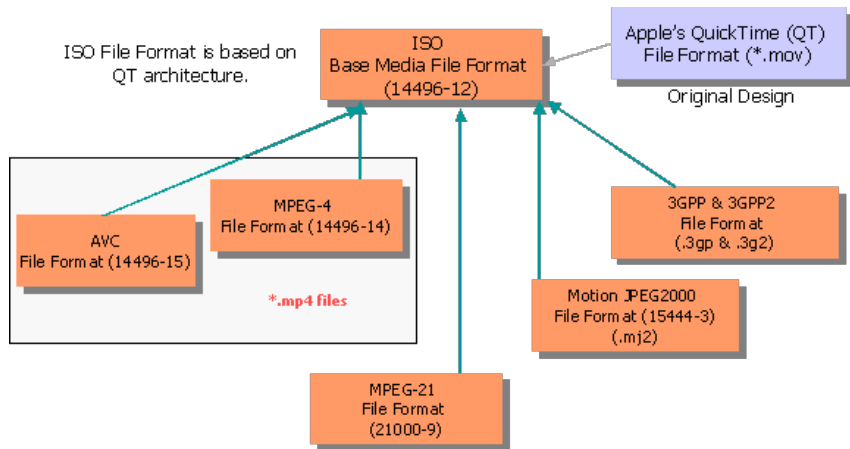
# Aktuelle Containerformate (Auswahl für Audio und Video)

| Containerformat                                   | Dateiendung |
|---|-------------|
| Advanced Systems Format (ASF)                     | asf         |
| Audio Video Interleave (AVI)                      | avi         |
| Blu-ray Disc Audio-Visual MPEG-2 Transport Stream | m2ts        |
| DivX Media Format                                 | divx        |
| Flash Video (FLV)                                 | flv         |
| Matroska  | mkv         |
| MPEG Video File                                   | mpg         |
| <b>MP4</b>  | <b>mp4</b>  |
| Ogg   | ogg         |
| QuickTime File Format                             | mov         |

- Informationen zur Audio- und Videoformatunterstützung: [http://en.wikipedia.org/wiki/Comparison\\_of\\_container\\_formats](http://en.wikipedia.org/wiki/Comparison_of_container_formats)

# Überblick MP4

- MP4: MPEG-4 Part 14 (ISO 14496-14)
- Ursprünglich von QuickTime File Format abgeleitet



Quelle: <http://www.tansee.com/what-is-mp4-video.html>

- Kleinste Einheit: Box (früher Atom genannt)
- MP4-Datei ist Sequenz von Boxes
- Boxes können weitere Boxes enthalten

→ Hierarchischer Aufbau

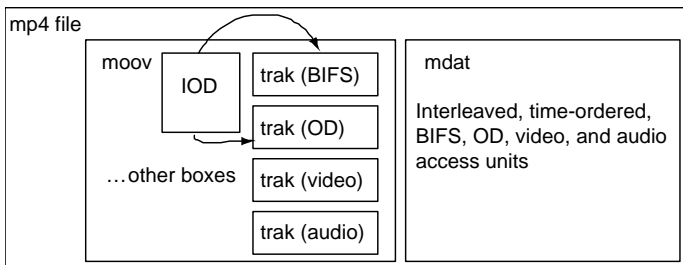
- Aufbau einer Box:
  - Optionaler Versionsheader
  - Typ (ID aus 4 Zeichen oder UUID (Universally Unique Identifier))
  - Länge (32 oder 64 Bit)
  - Inhalt (möglicherweise weitere Boxes)
- Obligatorische und optionale Boxes (Auswahl):
  - *ftyp*: Dateitypinformationen (ermöglicht Rückschluss auf Inhalt)
  - *moov*: Metadatencontainer
  - *mdat*: Container für Daten (Payload)
  - *free*: Unbenutzt

- *moov*-Boxstruktur:

- Initial Object Descriptor (*iods*): Datenstrom- und Dekodierinformation
- *mvhd*: Metainformationen wie Dauer und Erstellungszeitpunkt
- *trak*: Container für Metainformationen zu einer Spur (z.B. Video)

- *trak*-Boxstruktur:

- *tkhd*: Metainformationen wie Dauer und Erstellungszeitpunkt
- *mdia*: Container für spurspezifische Informationen

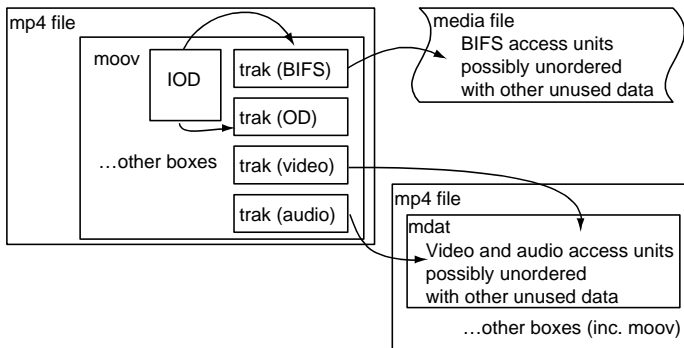


Quelle: ISO/IEC: Information technology – Coding of audio-visual objects – Part 14: MP4 file format, ISO/IEC 14496-14, 2003.



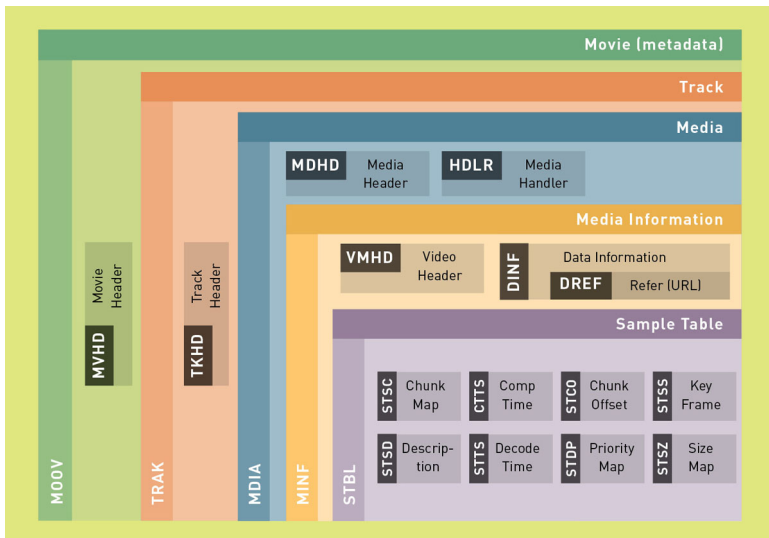
# Dateiaufbau III

- Eigentliche Daten (z.B. Videoframes) können in anderen Dateien (auch auf anderen Rechnern) liegen (Anwendungsfall Bearbeitung)
- Daten müssen nicht geordnet sein
- Daten müssen nicht verschachtelt (englisch *interleaved*) sein



Quelle: ISO/IEC: Information technology – Coding of audio-visual objects – Part 14: MP4 file format, ISO/IEC 14496-14, 2003.

# Typische *moov*-Boxstruktur



Quelle: Brown, M.: „White Paper: Media Container File Formats“.

[http://www.maximumpc.com/article/features/white\\_paper\\_media\\_container\\_file\\_formats](http://www.maximumpc.com/article/features/white_paper_media_container_file_formats) (8.9.2013), 2009.

# Dateitypidentifikation durch *ftyp*-Box

- *ftyp*-Box erlaubt Rückschluss auf Dateiinhalt
- Identifikation durch so genannte Brands (inkl. optionale Angabe kompatibler Brands)
- Brandregistrierung bei MP4 Registration Authority:  
<http://www.mp4ra.org/filetype.html>
- Häufig verwendete Brands (Auswahl):

| <b>ftyp-Typ (Code)</b> | <b>Dateiformat</b>        |
|------------------------|---------------------------|
| <i>avc1</i>            | H.264 nach MPEG-4 Part 15 |
| <i>isom</i>            | MPEG-4-Part-12-kompatibel |
| <i>m4a</i>             | iTunes Audio nach MPEG-4  |
| <i>mjp2</i>            | Motion JPEG 2000          |
| <i>mp42</i>            | MP4 (Version 2)           |
| <i>qt</i>              | QuickTime                 |

- *mvhd*-Box
  - Enthält `timescale`- und `duration`-Syntaxelement
  - `timescale` definiert Zeiteinheit (z.B. `timescale = 25` → 40 ms)
  - `duration` spezifiziert Dauer in Zeiteinheiten
- *tkhd*-Box
  - Enthält `duration`-Syntaxelement
  - `duration` spezifiziert Dauer in *mvhd*-`timescale`-Zeiteinheiten
- *mdhd*-Box
  - Enthält `timescale`- und `duration`-Syntaxelement
  - `timescale` definiert Zeiteinheit (wie in *mvhd*-Box)
  - `duration` spezifiziert Dauer in Zeiteinheiten
- *stbl*-Box erlaubt Zuordnungen zwischen Daten und Zeitangaben
  - Mehrere Unter-Boxes mit verschiedenen Informationen
  - Kleinste Dateneinheit: Sample (z.B. ein Frame)

# Zeitangaben II

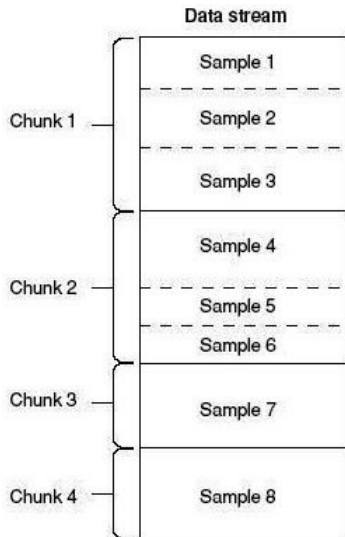
- Unterscheidung zwischen *Decoding Time* (DT) und *Composition Time* (CT) – Dekodier- vs. Darstellungszeitpunkt
- Zuordnung Samplenummer zu DT in *stts*-Box
- Zuordnung Samplenummer zu CT in *ctts*-Box (*CT to sample*)
- Speicherung der Werte als Differenz (Delta) zum jeweiligen Vorgänger
- Alle Angaben in `timescale`-Einheiten

| GOP                | /-- | --- | --- | --- | --- | --- | --\ | /-- | --- | --- | --- | --- | --- | --\ |
|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                    | I1  | P4  | B2  | B3  | P7  | B5  | B6  | I8  | P11 | B9  | B10 | P14 | B12 | B13 |
| DT                 | 0   | 10  | 20  | 30  | 40  | 50  | 60  | 70  | 80  | 90  | 100 | 110 | 120 | 130 |
| CT                 | 10  | 40  | 20  | 30  | 70  | 50  | 60  | 80  | 110 | 90  | 100 | 140 | 120 | 130 |
| Decode delta       | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  | 10  |
| Composition offset | 10  | 30  | 0   | 0   | 30  | 0   | 0   | 10  | 30  | 0   | 0   | 30  | 0   | 0   |

Quelle: ISO/IEC: Information technology – Coding of audio-visual objects – Part 12: ISO base media file format, ISO/IEC 14496-14, 2003.

- Daten in *mdat*-Box sind in Form von Chunks gespeichert
- Chunk: Aufeinanderfolgende Samples
- *stsc*-Box (*sample-to-chunk*): speichert
  - Erste Samplenummer jedes Chunks
  - Anzahl der Samples jedes Chunks
- *stco*-Box (*chunk offset*): speichert Byte-Offset jedes Chunks
- *stsz*-Box (*size*): Beschreibt Größen der einzelnen Samples (in Byte)
- Komplexere Zuordnungen möglich (ohne Details):
  - Mehrere Datenfragmente über *Track Fragments* (*traf*-Boxes)
  - Mehrere Spurfragmente über *Movie Fragments* (*moof*-Boxes)
  - *Timelines* über *Edit Lists* (*edts*-Box)

# Samplezuordnung II

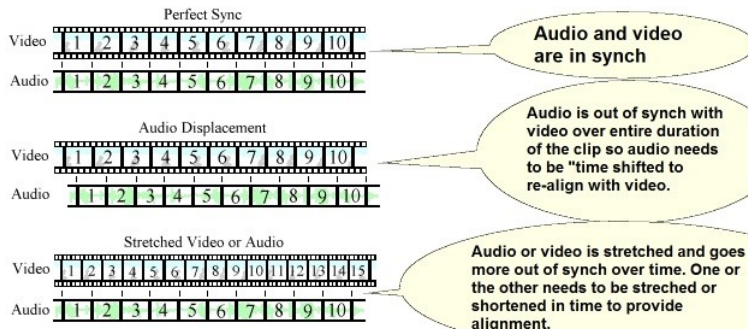


Quelle: <http://blog.csdn.net/tx3344/article/details/8506131>

- Samplenummer kann CT zugeordnet werden
- Synchronisation der Spuren durch CT möglich
- Umrechnung der CTs mehrerer Spuren eventuell notwendig (bei unterschiedlicher *timescale* entsprechend umzurechnen)
- Synchronisationsherausforderungen:
  - Mensch ist empfindlich auf Audio-/Video-Asynchronität (filmmaterialabhängig, allgemein ca.  $\pm 22$  ms tolerierbar)
  - Unterschiedliche Sampledauer
  - Eventuell separate DT-Berücksichtigung (und -umrechnung) notwendig
  - Synchronisation durch mögliche Filter nach der Dekodierung schwierig
  - Signallaufzeiten in Hardware schwer zu berücksichtigen (separate Hardwareunterstützung notwendig)
  - Physikalische Signallaufzeit nicht beeinflussbar



# Samplezuordnung IV



Quelle: <http://www.magix.info/us/how-to-make-very-fine-audio-adjustments-to-sync.knowledge.892859.html>

- Menschliche Asynchronitätstoleranz sehr niedrig

- *stss*-Box (*sync-sample*): Speichert Liste von Samplenummern, ab denen Dekodierung möglich ist (Prädiktionsgrenze)
- *stsh*-Box (*shadow-sync*)
  - Speichert alternative Samplenummer für einige der (oder alle) Samplenummern der dazugehörigen *stss*-Box
  - Alternative Samplenummer kann ebenfalls genutzt werden
- *sdtb*-Box (Erweiterung für H.264): speichert Abhängigkeiten jedes Samples (feingranulare Version der *stss*-Box-Daten)
  - Abhängigkeit eines Samples von anderen (ja/nein/weiß nicht)
  - Abhängigkeit anderer Samples von einem Sample (ja/nein/weiß nicht)
  - Sample redundant (ja/nein/weiß nicht)
- Weitere Möglichkeiten (ohne Details)
  - Bei mehreren Datenfragmenten über *tfra*-Boxes
  - Bei mehreren Spurfragmenten über *mfra*-Boxes

- *stsd*-Box in *stbl*-Box: Sample-Beschreibung
  - Bildauflösung in Pixel (bei Videospur)
  - Kanalanzahl (bei Audiospur)
  - Anzahl Frames pro Sample (typischerweise einer)
  - Name (ID) des verwendeten Kompressionsalgorithmus
- *udta*-Box: Container für Zusatzinformationen
  - Kann in *trak*- oder *moov*-Box beinhaltet sein
  - Entsprechende Zugehörigkeit des Inhaltes
    - *cpri*-Box: Copyright-Informationen
- *meta*-Box: Container für weitere Zusatzinformationen (in Datei, *trak*- oder *moov*-Box beinhaltet)
- Liste aller registrierten Boxes: <http://www.mp4ra.org/atoms.html>

Fragen?