

Drop Frame Timecode and Fractional Frame Rates

Lessons from the past... and looking to the future

John Pallett, Telestream

Bruce Devlin, Dalet

Let's start with the elephant in the room...



Common Perceptions about DF vs. NDF

Generally people don't understand why we have fractional frame rates

00:00:00:00

30 fps



Colon!

00:00:00;00

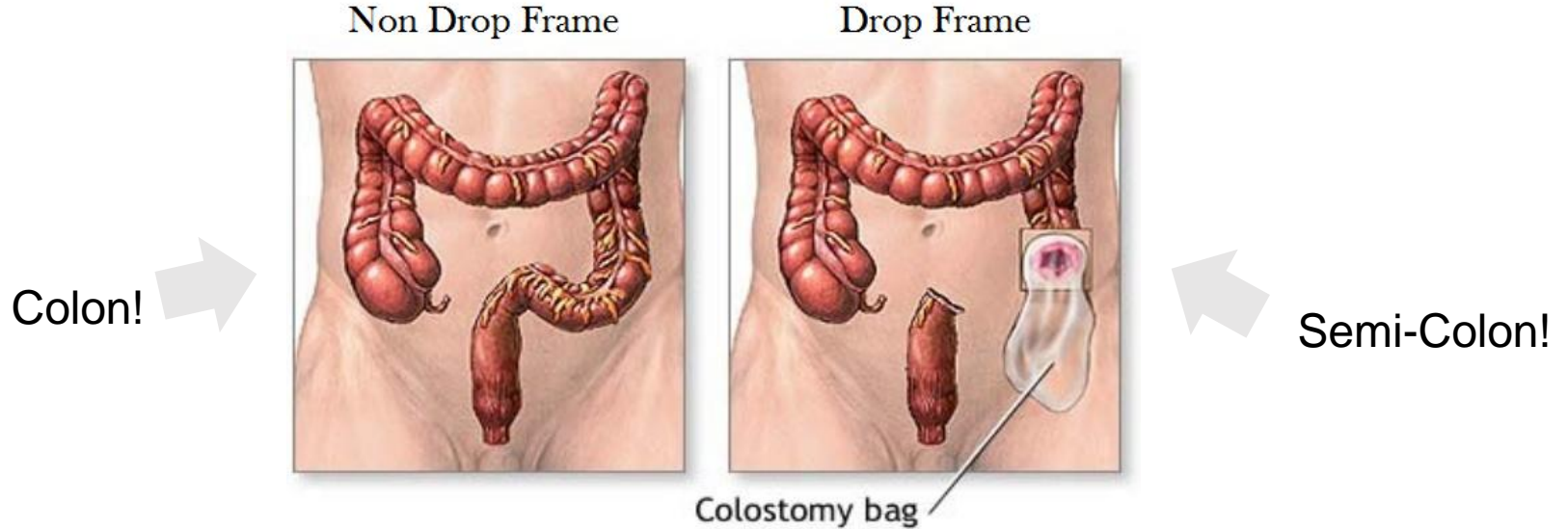
29.97 fps



Semi-Colon!

Why on earth did we do this? It just means s*#@ will fall on the floor!

Common Perceptions about DF vs. NDF



Why on earth did we do this? It just means s*#@ will fall on the floor!

What we will talk about

- Why fractional frame rates exists... and why they will continue to exist
- Problems from the past... and how people solved them
- Looking to the future

How did we get here?



Why 29.97?

- Black and white television was 30 frames per second, with luminance (AM) and audio (FM) at a fixed frequency distance (4.5 MHz)
- Adding the color carrier frequency required inserting a third band...
 - *Without causing artifacts on the existing black and white TV sets*
 - *So, color carrier frequency needed to be an odd harmonic of half line frequency*
 - *Some math was done...*

Ratio of Horizontal line rate change = 1.001 : 1, 30fps / 1.001 = 29.97

“It’s Backwards Compatible!”

Why 29.9700299700299700299700299700299700...?

- Black and white television was 30.0000000000 frames per second, with luminance (AM) and audio (FM) at a fixed frequency distance (4.5 MHz)
- Adding the color carrier frequency required inserting a third band...
 - Without causing artifacts on the existing black and white TV sets
 - So, color carrier frequency needed to be an odd harmonic of half line frequency
 - Some math was done... *and this solution seemed the best compromise!*

Ratio of Horizontal line rate change = 1.001 : 1, 30fps / 1.001 = 29.97

“It was a Backwards Compatible Solution!”

Why is timecode important?

Timecode indicates the timing of each frame. It is used to ensure:

- **Alignment** between independent devices (e.g. video and audio recorders)
- **Alignment** between assets (e.g. captions and video)
- Correct **edit** points (e.g. cutting at the correct time)
- Accurate playback **duration**

The problem with 29.97 time code...

“It’s just like leap years”

Your video is running at 29.97 frames per second

If you count to 30 for every second, you will be off by **108 frames per hour**

This is 3.6 seconds per hour, or roughly **2 minutes every day**

*Congress refused to change the standard duration of a minute
and refused to speed up the rotation of the earth, so...*

A new time code scheme was invented to account for the difference

Teaching your child to count the DF way

“It’s just like leap years”

- Duration of a solar year is slightly less than **365.25 days**
- We have February 29th when the year number is a factor of 4
 - Except on centuries...
 - ... that are not multiples of 400

Example:

- 1701 and 1700 and were **not** leap years
- 1996 and 2000 **were** leap years

Teaching your child to count the DF way

“It’s just like leap years”

- Duration of a 29.97fps hour (at 30fps) is 108 frames too long
- So, we **skip two frames every minute**
 - Drop-frame timecode skips ;00 and ;01
 - ... except on multiples of ten minutes

Example:

- 00:05:59;28 – 00:05:59;29 – **skip two** – 00:06:00;02 – 00:06:00;03
- 00:09:59;28 – 00:09:59;29 – 00:10:00;00 – 00:10:00;01

War Stories...



When timecode & frame rates go wrong...

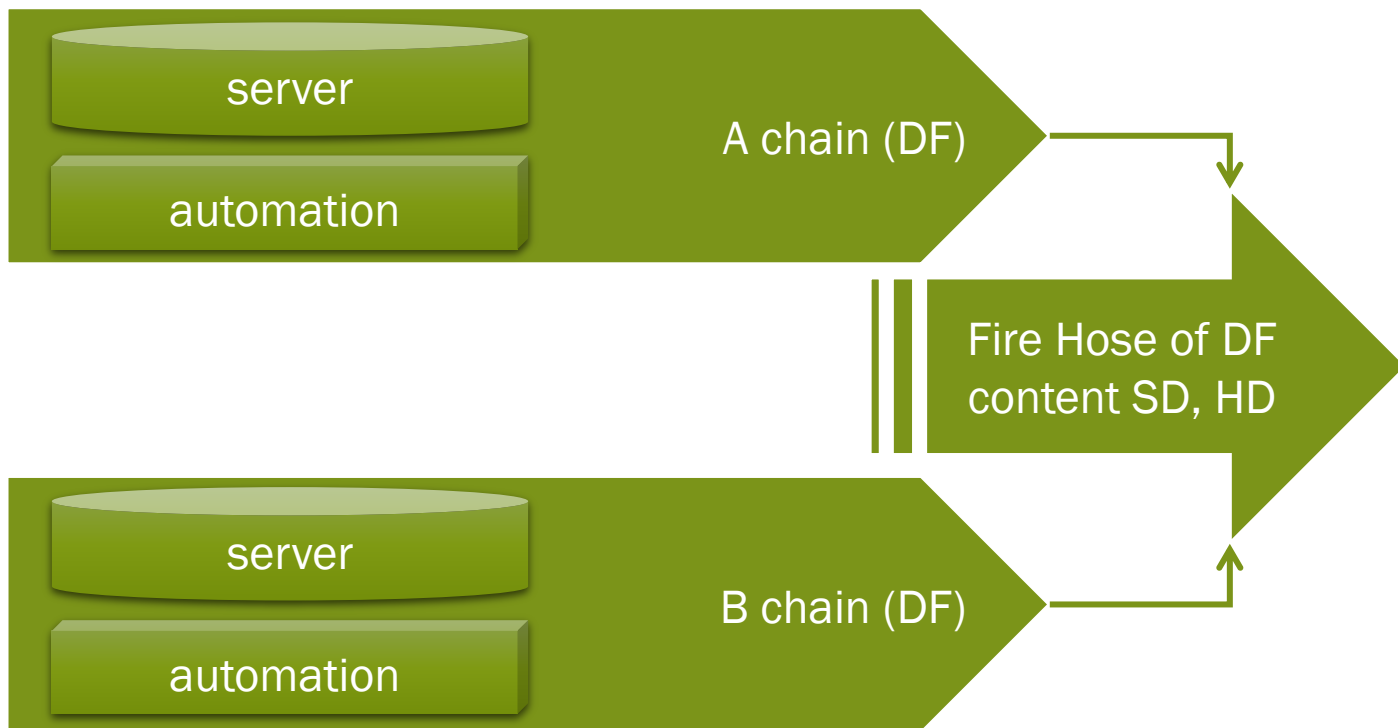
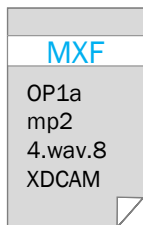
- Media is in the “wrong” time base
- Systems are not using the same time base
- The time base metadata is lost or unavailable (e.g. SCC files)

In other words... things go wrong when either...

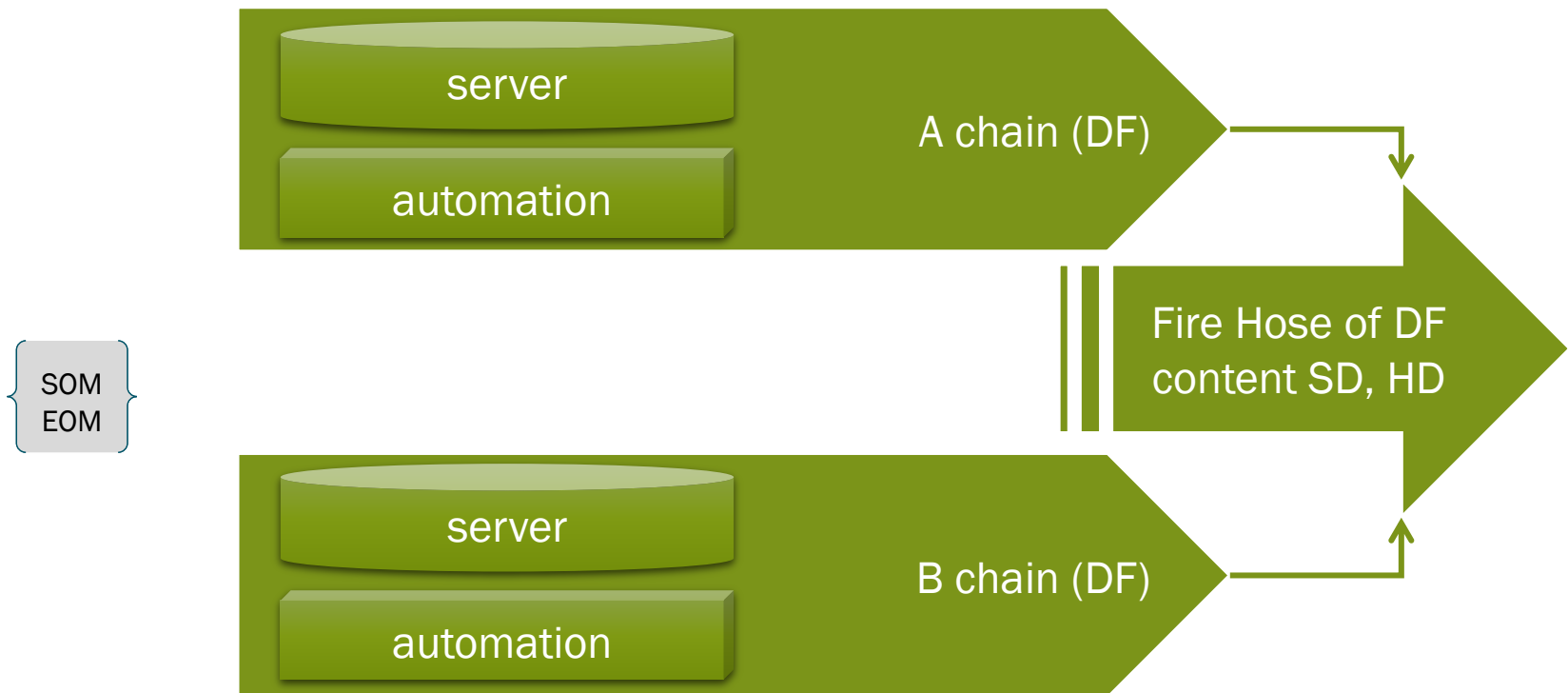
you don't know the time base (and guess wrong)

or a system is using the wrong time base

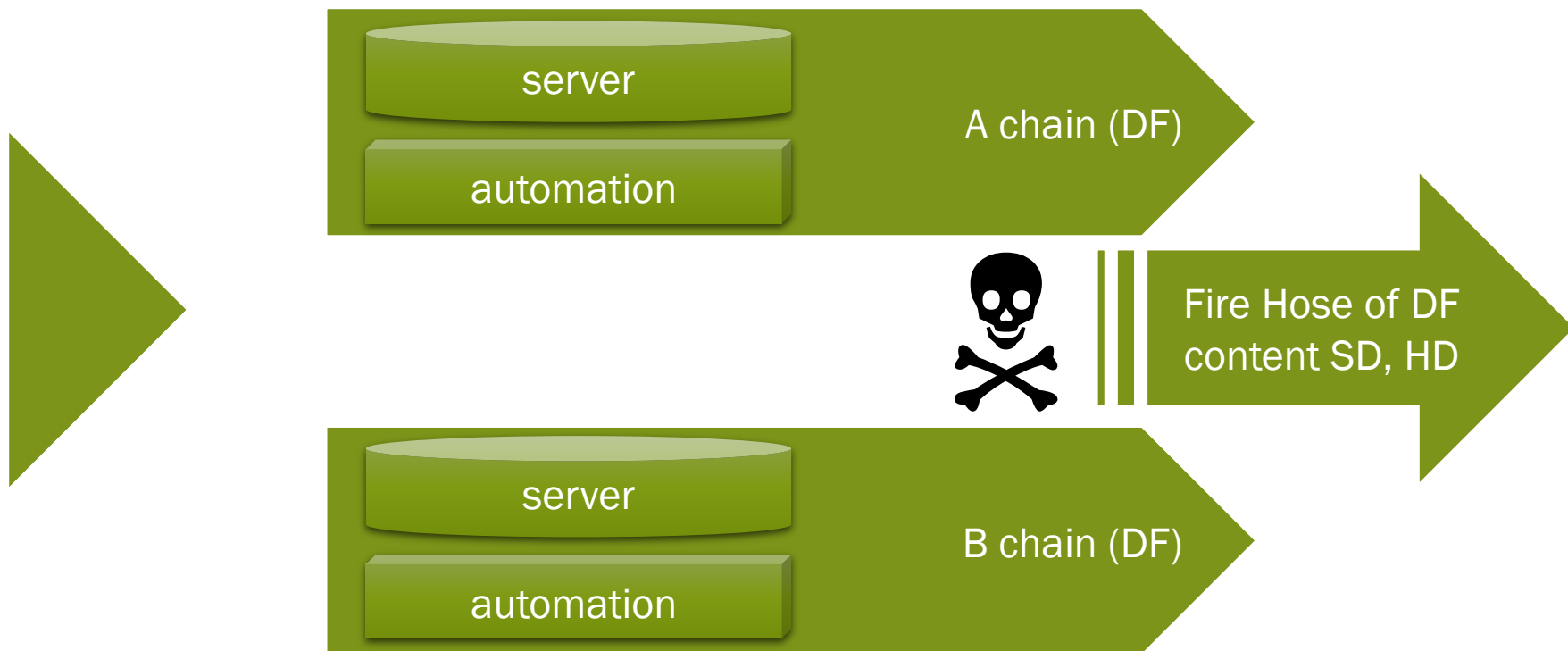
Example: The “I didn’t think of that” problem



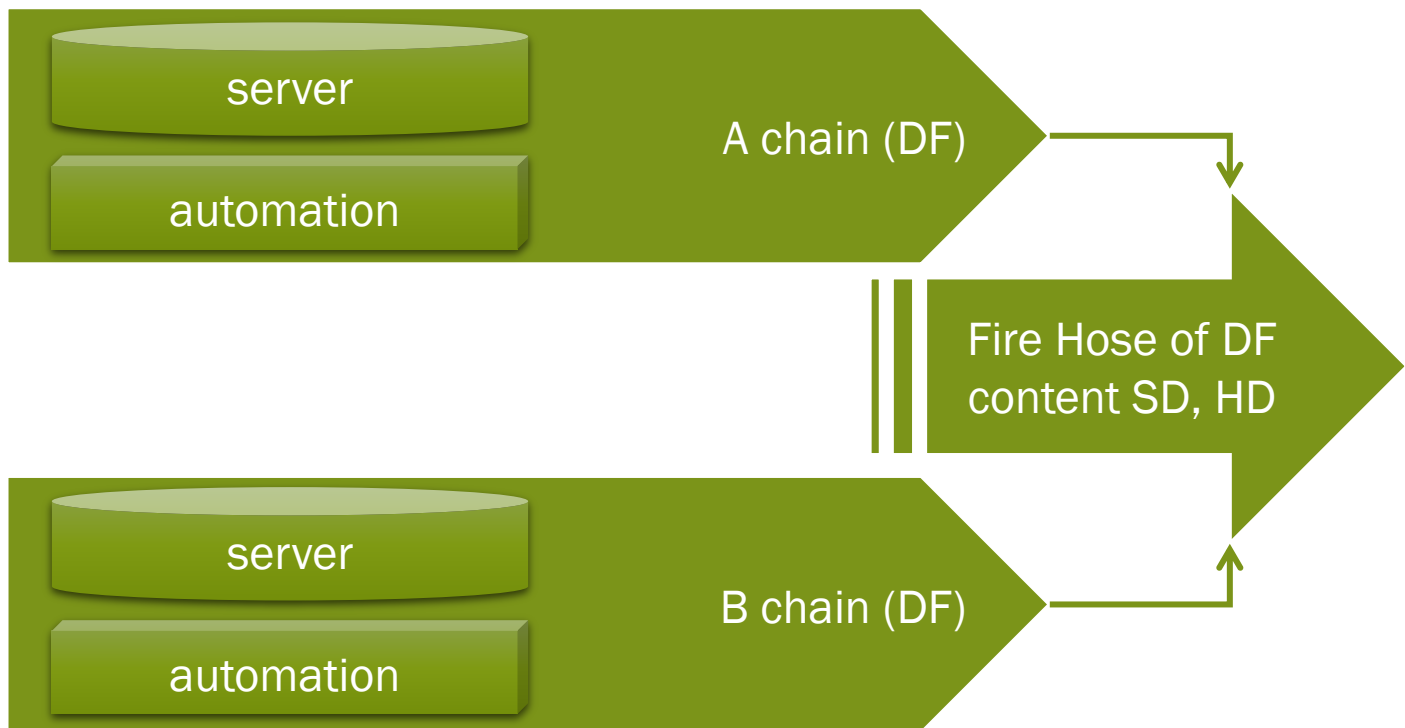
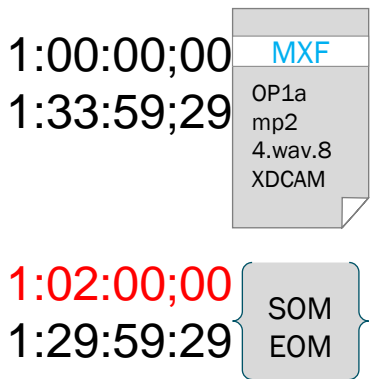
Example: The “I didn’t think of that” problem



Example: The “I didn’t think of that” problem



Example: The “I didn’t think of that” problem



Example: The “I don’t know what I’m getting” problem

- Netflix accepts thousands of submissions each month
- Part of the package is captioning, often in an **SCC** file
- If the video is at 23.976 and the captions are at 24... captions drift

- This was the #1 QC problem for some time
- About a 4% failure rate

NETFLIX

Example: The “I don’t know what I’m getting” problem

- SCC files do not have a way of specifying the time base
 - And there is no standardized way to do 23.976 DF timecode

GOOD NEWS: Netflix converts everything to **media time**...

... so if they know what it is, they can get it right

NETFLIX

Example: The “I don’t know what I’m getting” problem

A two-part solution

- **Better Information:** New field in the Netflix manifest file for caption time base
- **Broader support:** They also allow caption files with media time

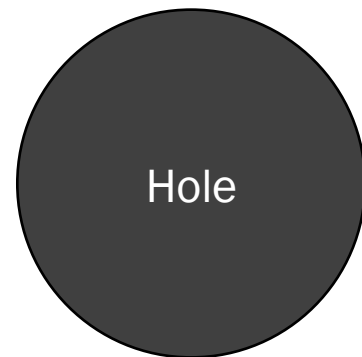
RESULT: Error rate dropped from 4% to less than 1%

... and usually an XML tweak fixes things!

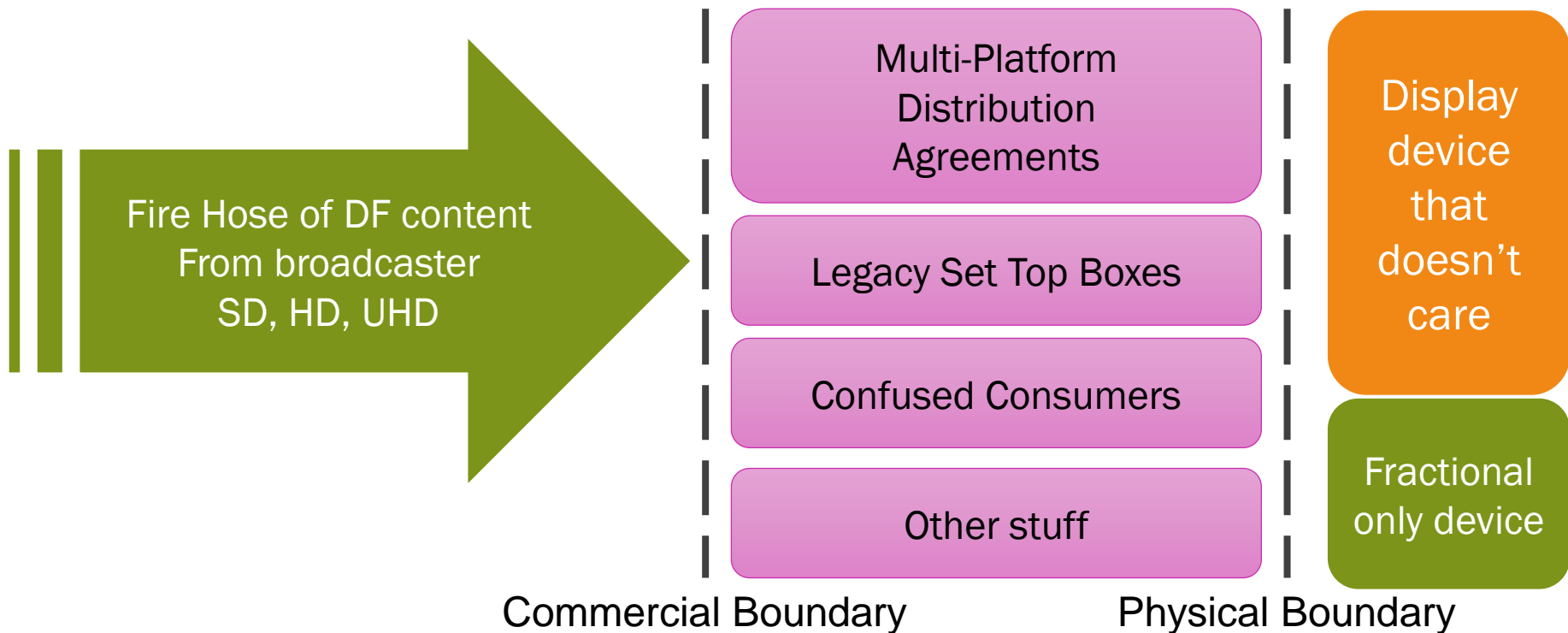
NETFLIX

Example: The “I don’t know what I have” problem

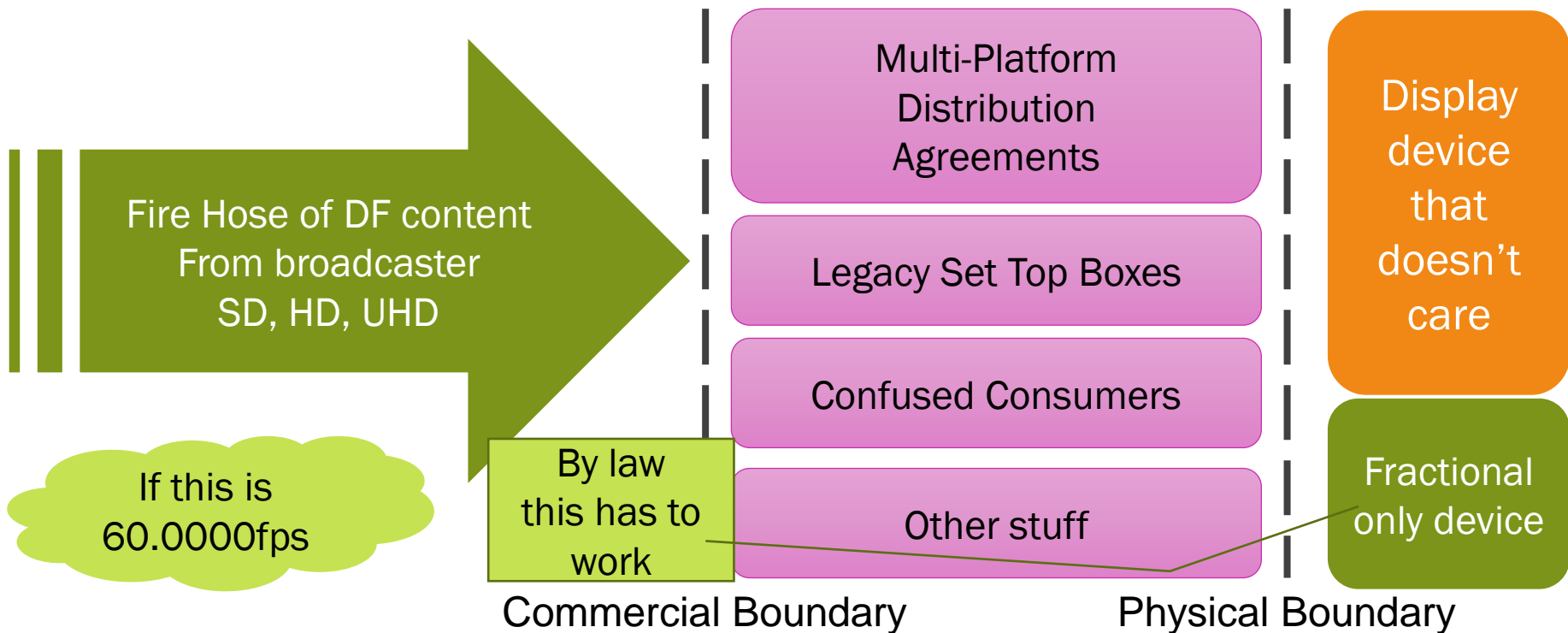
- Imagine a world ...
- Captions had be created, QC’d and aligned manually
- Time to digitize the archive
 - Some caption originals existed
 - Some were 23.98
 - Some were 29.97
 - Some were 30.0
 - Some were aligned to 1st frame
 - Some were aligned to start of clock
 - Some were aligned to the moon (or maybe the tides)



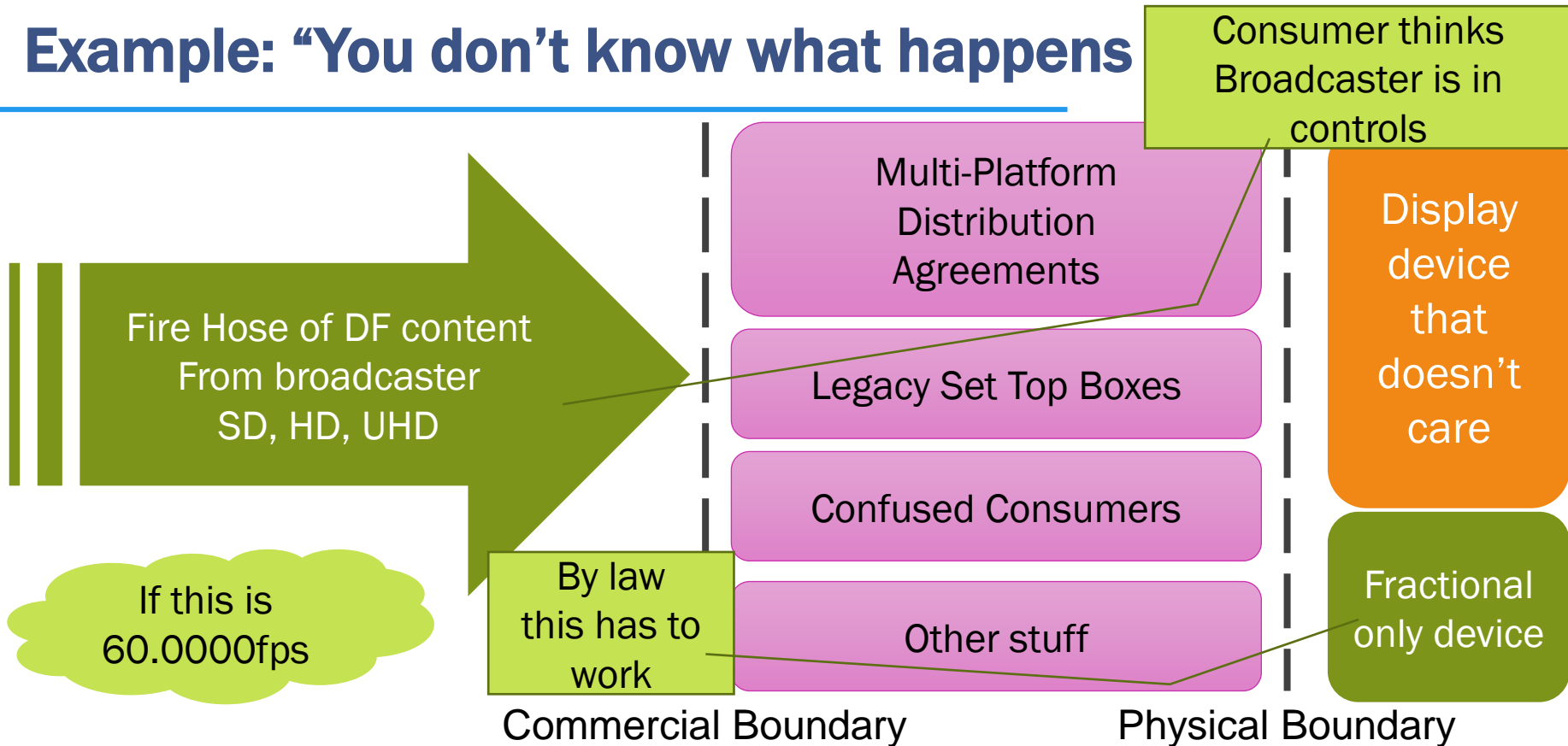
Example: “You don’t know what happens downstream”



Example: “You don’t know what happens downstream”



Example: "You don't know what happens"



Other Challenges with DF



Converting Frame Rates

Internationalization

- 29.97 to 25

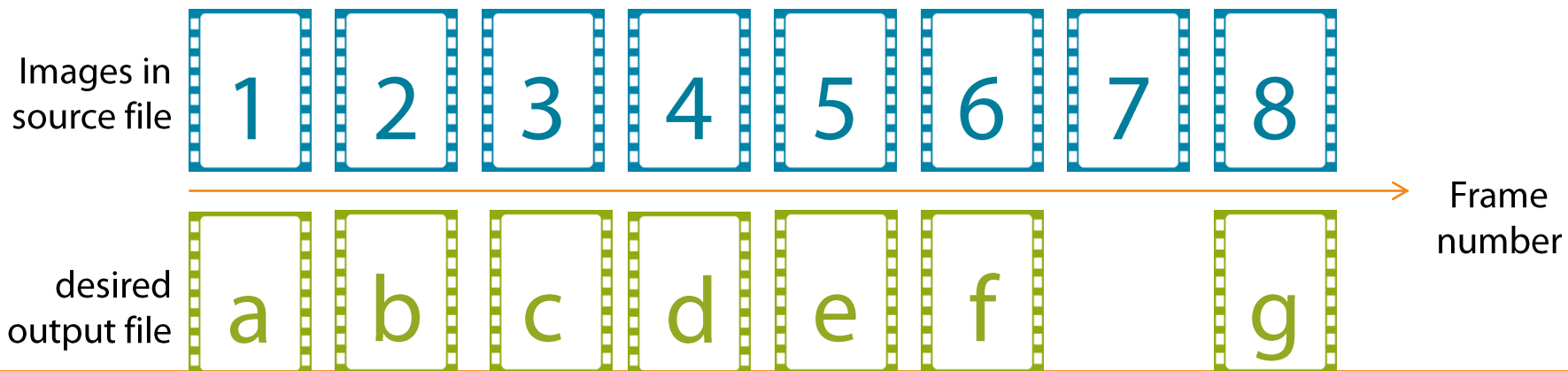
Production to broadcast

- 30 to 29.97
- 24 to 29.97
- 24 to 23.976

→ 23.976 does *not* have a standard for drop frame timecode

High quality transcoding solution can handle these situations with files

Frame Rate Conversion Can Be Solved



Conversions – what and where

- Process video to maintain duration?
- Process audio & metadata to match video frame count duration?
- 2:3 inserting has served the DF community well for decades
- For 120fps - why not frame double and hope quality is good enough?
- For 120fps - why not frame halve and hope quality is good enough?
- What are the trade offs? processing time vs. video quality?
- Are captions and ancillary data be preserved?
- Is that even relevant?

in	out	23.98PsF	24p	25i	29.97i	30p	50p	59.94p
23.98PsF	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> audio resample 	<ul style="list-style-type: none"> speed-up interpolate 	<ul style="list-style-type: none"> 2:3 insert interpolate 	<ul style="list-style-type: none"> 2:3 insert interpolate 	<ul style="list-style-type: none"> speed-up interpolate 	<ul style="list-style-type: none"> 2:3 insert interpolate 	
24p	<ul style="list-style-type: none"> audio resample 	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> speed-up reinterlace 	<ul style="list-style-type: none"> 2:3 insert interpolate 	<ul style="list-style-type: none"> 2:3 insert interpolate 	<ul style="list-style-type: none"> frame rpt 	<ul style="list-style-type: none"> interpolate 	
25i	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> frame rpt 	<ul style="list-style-type: none"> interpolate 	
29.97i	<ul style="list-style-type: none"> 2:3 remove interpolate 	<ul style="list-style-type: none"> 2:3 remove Interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> audio resample 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> frame rpt 	
30p	<ul style="list-style-type: none"> 2:3 remove Interpolate 	<ul style="list-style-type: none"> 2:3 remove interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> audio resample 	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> audio resample 	
50p	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> Interpolate 	<ul style="list-style-type: none"> field drop 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> pass-thru 	<ul style="list-style-type: none"> interpolate 	
59.94p	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> field drop 	<ul style="list-style-type: none"> audio resample 	<ul style="list-style-type: none"> interpolate 	<ul style="list-style-type: none"> pass-thru 	

Editing...

- Trim values may be in an incorrect time base
- EDL values may be incorrect as well
- Editing NDF assets into a long-form output may require timecode restriping
- Mixing DF and NDF assets requires care
 - EDLs may need to contain both types of timecode
 - Rendering engines must be aware of – and handle correctly – these cases

When You Mix Frame Rates & Timecode systems...

- You can generate system and operational problems



Magic Box
to fix mixed
DF & NDF
issues

The Future



DF timecode is here to stay

- Over **\$70 billion** is generated annually broadcasting using DF timecode
- Estimated that over **\$50 billion*** in equipment would have to be replaced in order to switch entirely to 30fps instead of 29.97
- This is not going to happen
- So, let's talk about how to make it work...

** Courtesy of Devoncroft Partners*

Lesson #1 – Standards Make Life Easier

- The lack of 23.976 drop frame does make things difficult
- Different solutions address this in different ways
 - The result may be **accurate duration**...
 - ... but how do you handle **editing** and **synchronization**?

DF Standard for 47.952 , 59.94, 119.88, and 239.76 fps would help...

... but as long as you know the asset time base, NDF can work (sort of)

Lesson #2 – Avoid Ambiguity

- Assets should *always* have an indication of their time base

Key example: IMF

- May contain edit points
 - May have multiple streams that need to be aligned
 - Will have content duration
-
- Can't we just use frame counts instead of timecodes?
 - Yes, but getting accurate duration for a complex EDL asset will be difficult
 - Are you sure your edit points are correct?

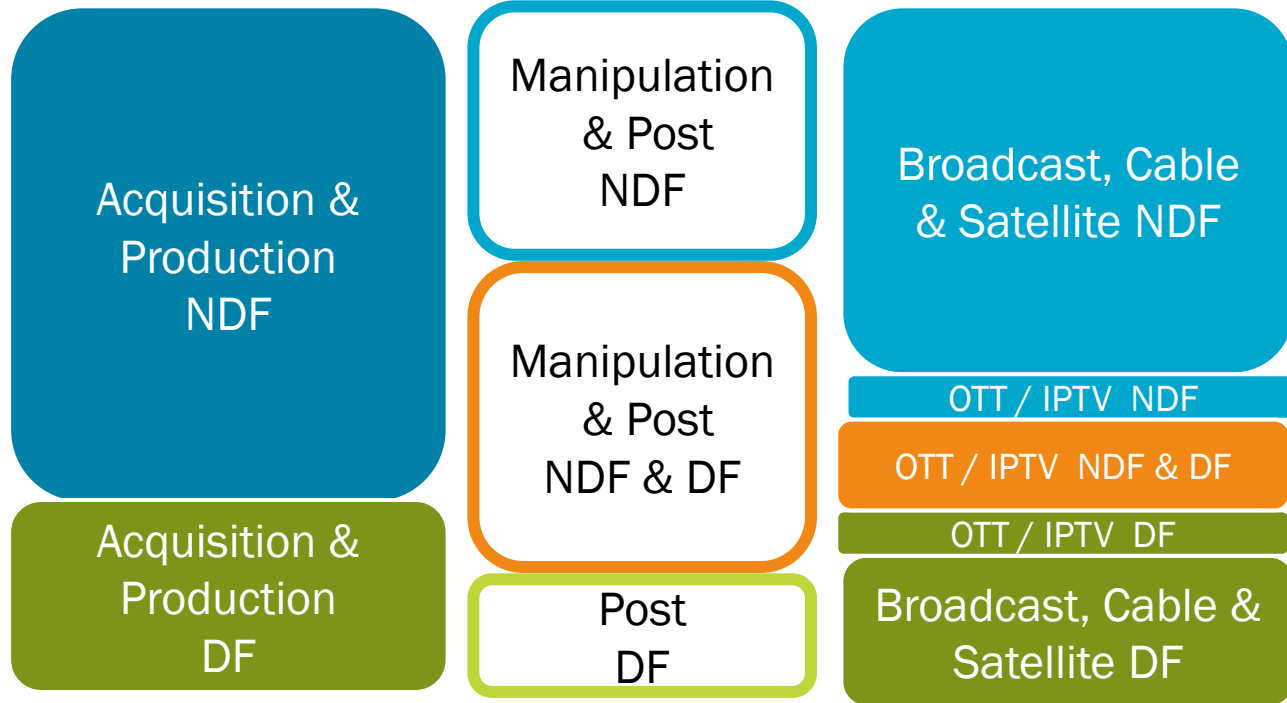
The short version on this is that every tool and asset in the IMF preparation chain will have to be consistent and correct

Problems that could go away

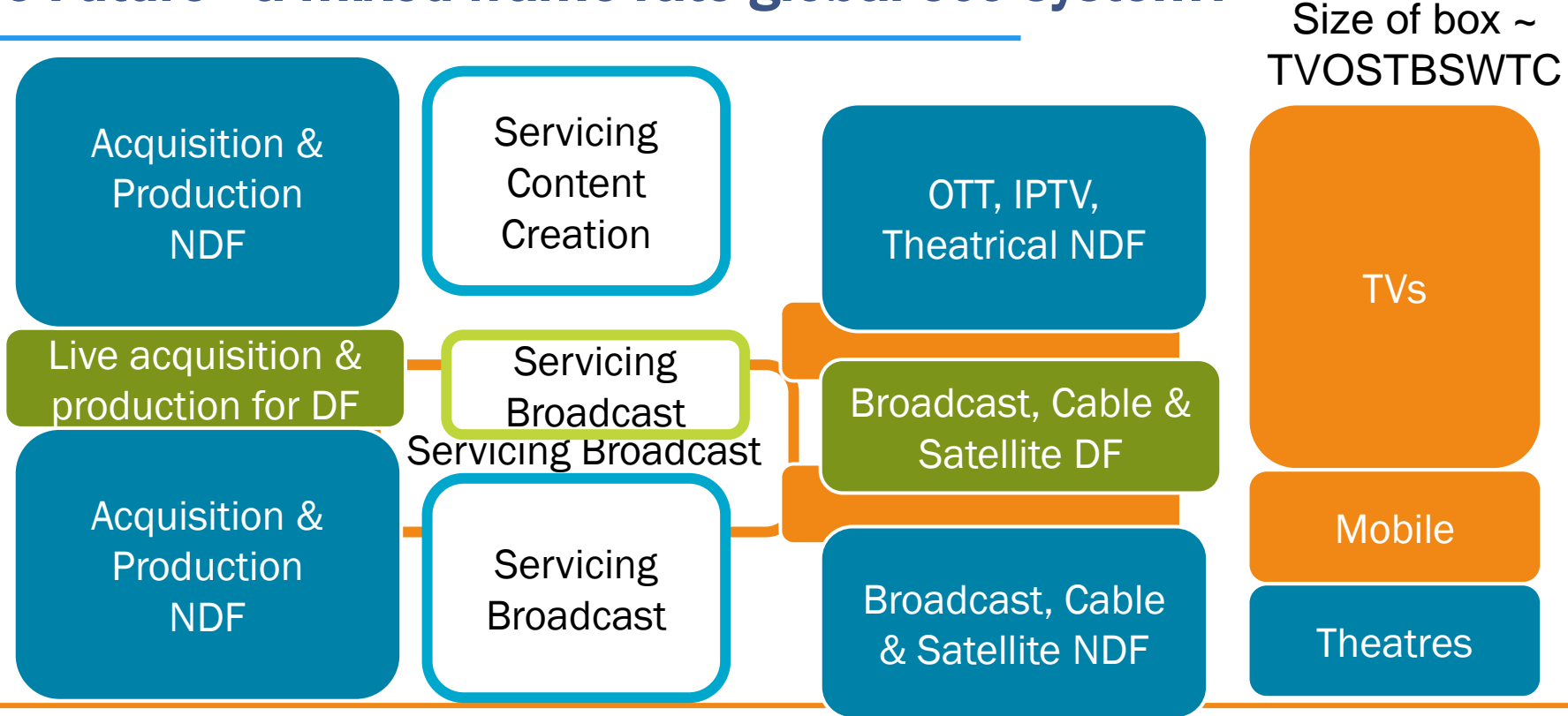
- With a little cooperation at the system level we could eliminate
 - Systems that don't talk with each other; they need to be aligned
 - Teams that are working on different time bases; they need flexibility
 - Assets that do not indicate their time base
 - With IMF a new challenge arises:
 - *The possibility of assets that are not consistent with themselves*
 - *As long as all element assets are annotated correctly this is probably OK*

The Future - a mixed frame rate global eco-system?

Size of box ~
TVOSTBSWTC



The Future - a mixed frame rate global eco-system?



In Summary



In summary

Production
Non Drop Frame

Boundary
Layers

Broadcast
Drop Frame

Issues tend to occur where **drop frame** and **non-drop frame** intersect

- ... if your systems are not configured correctly
- ... if you don't know what you have
- ... if you can't control the process
- ... if there is no standard

Our Analogy...

Hydrogen

Boundary
Layers

Oxygen

Manage the boundary and things will go well
Otherwise... bad things can happen
(... or at least things might drift)



Q&A

