Solutions for streaming video delivery

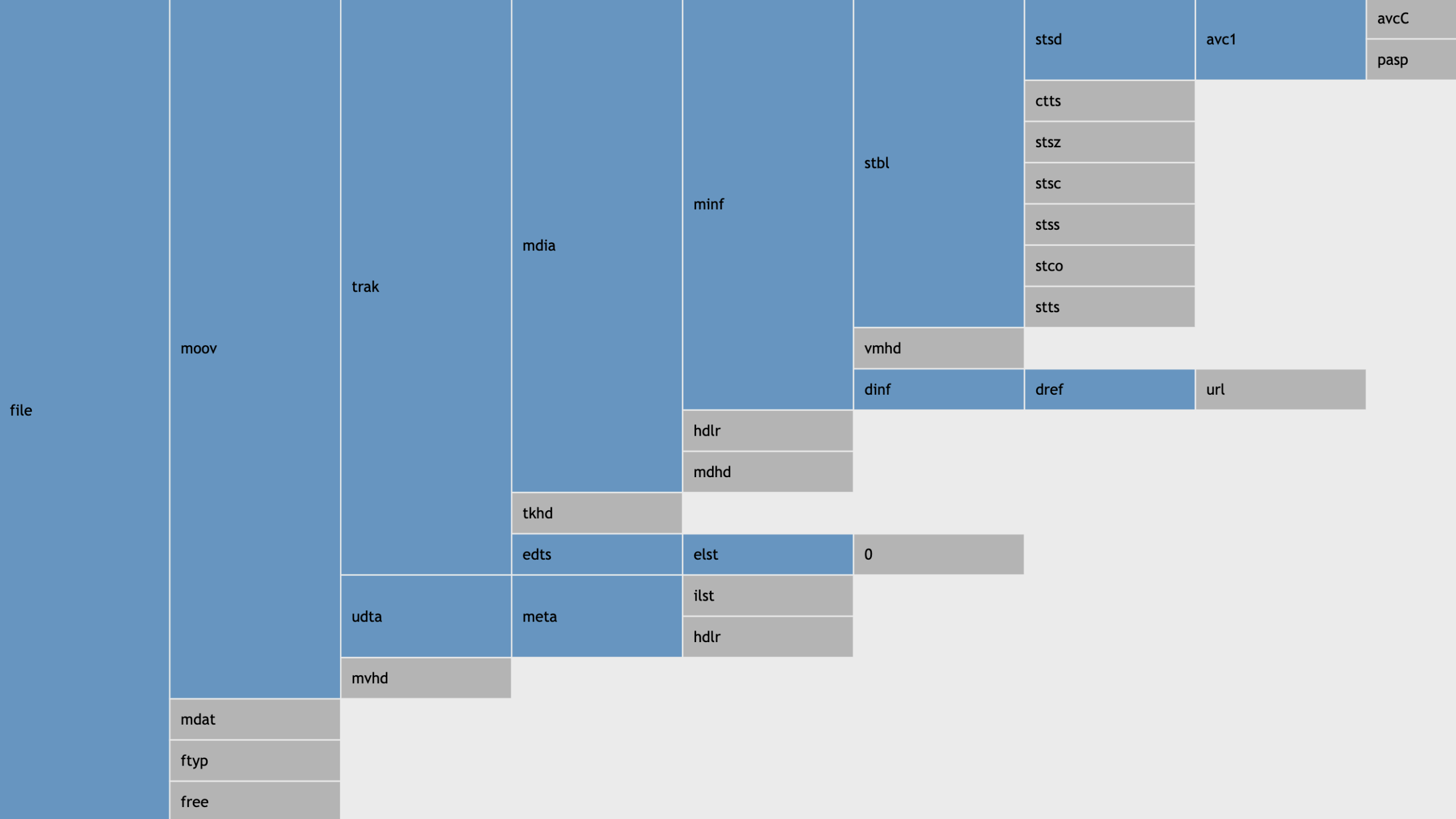
Boy van Dijk

Solutions engineer and technical writer

Unified Streaming

Optimizing MP4

For large scale, dynamically packaged, on-demand delivery of remotely stored content


file
moov
mdat
ftyp
free
trak
udta
mvhd
mdia
meta
tkhd
edts
minf
hdlr
mdhd
ilst
hdlr
elst
0
stbl
vmhd
dinf
dref
url
stsd
avc1
avcC
pasp
ctts
stsz
stsc
stss
stco
stts

mp4split

Boxes inside an MP4 where its ‘index’ can be stored

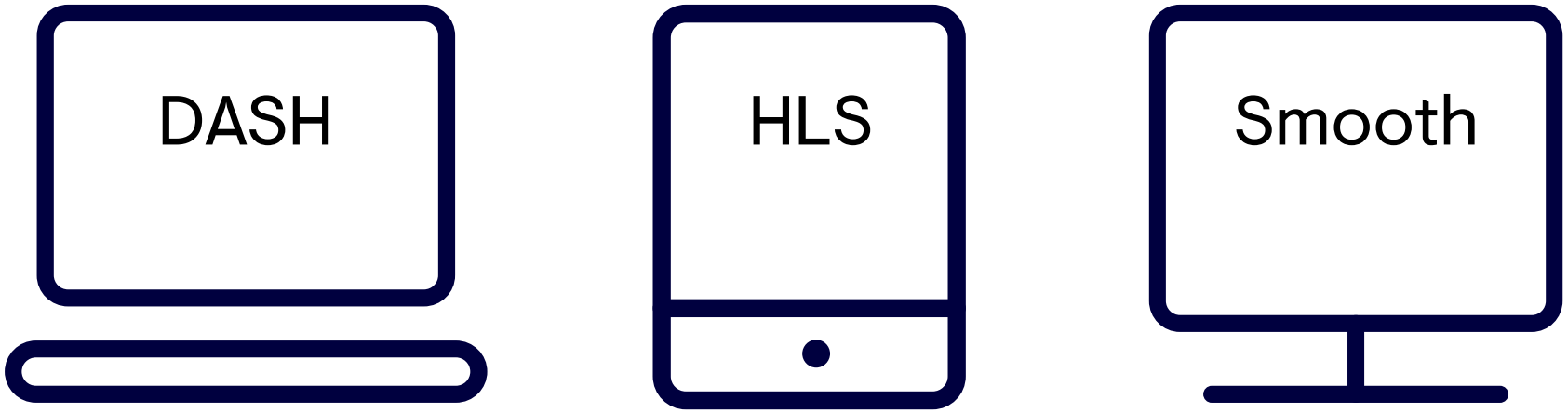
- ✓ Moov (progressive MP4)
- ✓ Mfra (Microsoft Smooth Streaming)
- ✓ Sidx (CMAF)

Middleman

dref MP4

data reference

Just-in-time packaging: from one source to all formats



Just-in-time packager

+

Remote storage

Unified Origin

Stream any format

So...

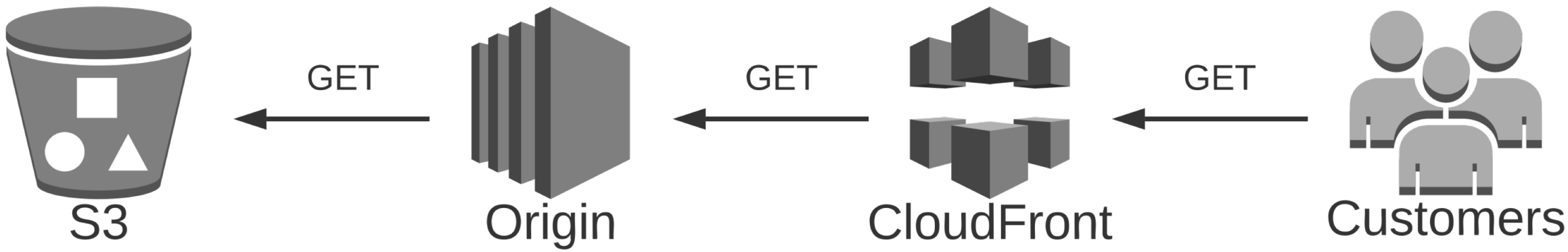
How do dref MP4's help?

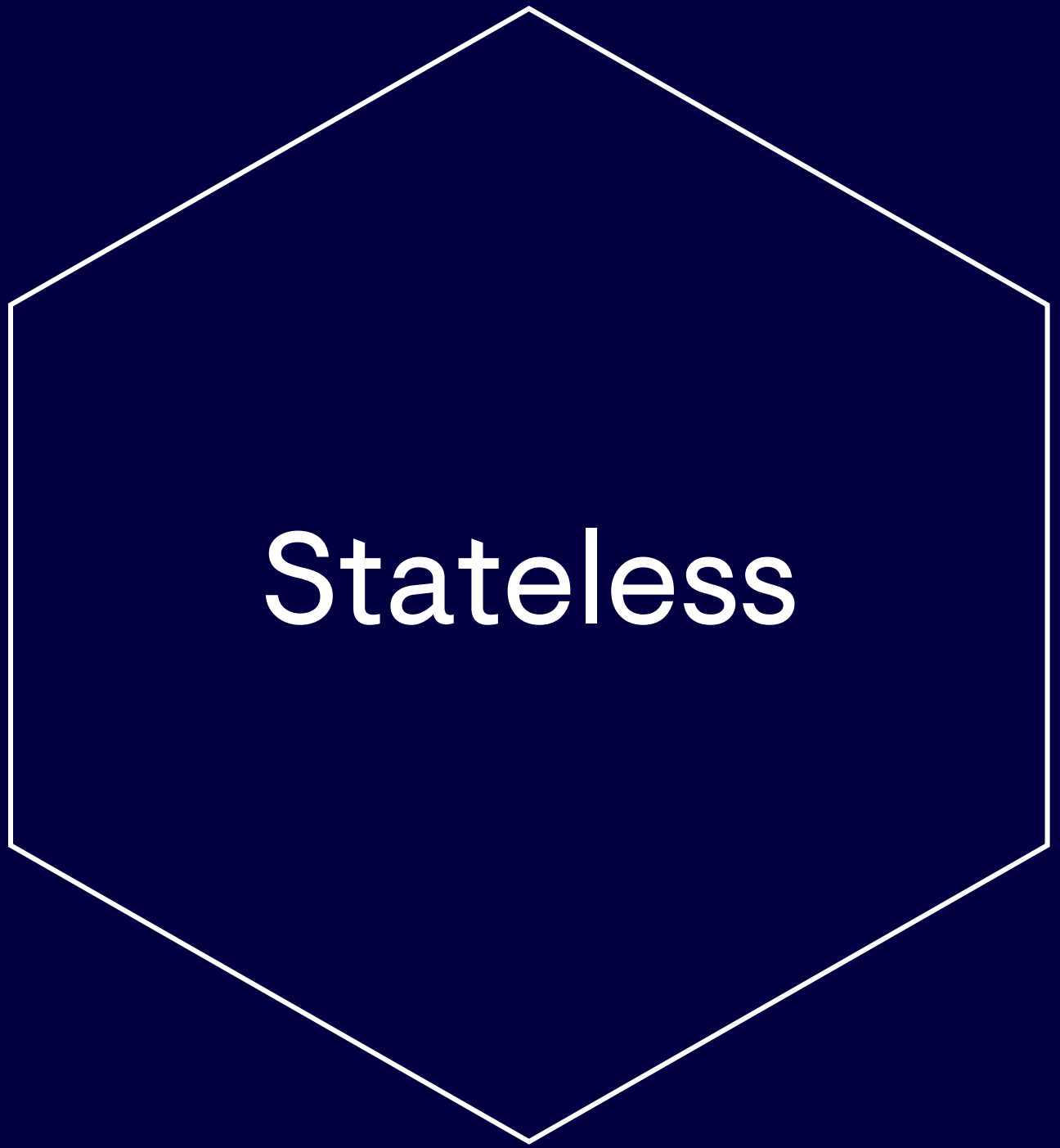
Latency

Remote storage of large video libraries

- ✓ Relatively cheap
- ✓ High throughput
- ✓ ‘Infinite’ storage

Longtail


S3
GET
Origin
GET
CloudFront
GET
Customers


Stateless

Solution:
Efficient caching

Remote source

- ▷ Contains media data
- ▷ Heavyweight
- ▷ Inefficient to cache

dref MP4

- ▷ References media data
- ▷ Lightweight
- ▷ Efficient to cache

Choose your own caching solution

- ✓ NGINX
- ✓ Apache
- ✓ Varnish
- ✓ Other

How small?
Tens of MB’s
for a full movie

Average request time (Unified Origin)

No caching

- ▷ MPD: ~1160 ms
- ▷ Init: ~184 ms
- ▷ Media: ~240 ms

Cached dref MP4

- ▷ MPD: ~16 ms
- ▷ Init: ~13 ms
- ▷ Media: ~160 ms

Other research:
+10 – 20% throughput

Finally: some of our other use cases for dref MP4's

- ✓ On-the-fly conversion of fragmented to progressive MP4 (Origin)
- ✓ On-the-fly content stitching (Remix)
- ✓ On-the-fly interleaving of Timed Metadata (Remix)

Feel free to ask me questions ☺

docs.unified-streaming.com
search for: ‘storage caching’

~

scientific presentation: tinyurl.com/drefmp4