ENCODING LIVE AND VOD FOR HEVC/HLS

A Joint SLC/RealEyes Production

Agenda

- Our assumptions and goals
- Section I: Introduction to HEVC
- Section II: Playback performance
- Section III: Introduction to HLS
- Section IV: Specification overview: HEVC in HLS
- Section V: Producing HEVC/HLS

Section V. Producing HEVC/HLS

- DIY – VOD
  - FFmpeg – create the A/V files
  - Bento4 – package and manifest files
- Third party alternatives
  - Live
  - VOD

Creating HEVC Files in FFmpeg

- Use the x265 codec
  - Widely recognized as one of the fastest and highest quality
  - Need to compile Main10-specific version
- All scaling and other syntaxes apply
- Need to choose profile and preset (unless defaults OK)
- Must use –x265-params command for some parameters

Encoding x265 in FFmpeg

```
ffmpeg -y -i TOS_1080p.mov -c:v libx265 -preset slow -x265-params profile=main:keyint=48:
min-keyint=48:scenecut=0:ref=5:bframes=3:b-adapt=2:bitrate=4000:vbv-maxrate=4400:vbv-bufsize=4000
-an -pass 1 -f mp4 NUL && \

ffmpeg -i TOS_1080p.mov -c:v libx265 -preset slow  -x265-params profile=main:keyint=48:
min-keyint=48:scenecut=0:ref=5:bframes=3:b-adapt=2:bitrate=4000:vbv-maxrate=4400:vbv-bufsize=4000
-an -pass 2 TOS_1080p_h.mp4

ffmpeg -i TOS_1080p.mov -c:v libx265 -s 1280x720 -preset slow  -x265-params profile=main:
keyint=48: min-keyint=48:scenecut=0:ref=5:bframes=3:b-adapt=2:bitrate=1000:vbv-maxrate=1100:
vbv-bufsize=1000 -an -pass 2 TOS_720p_l.mp4
```

- Integrate x265 commands into FFmpeg

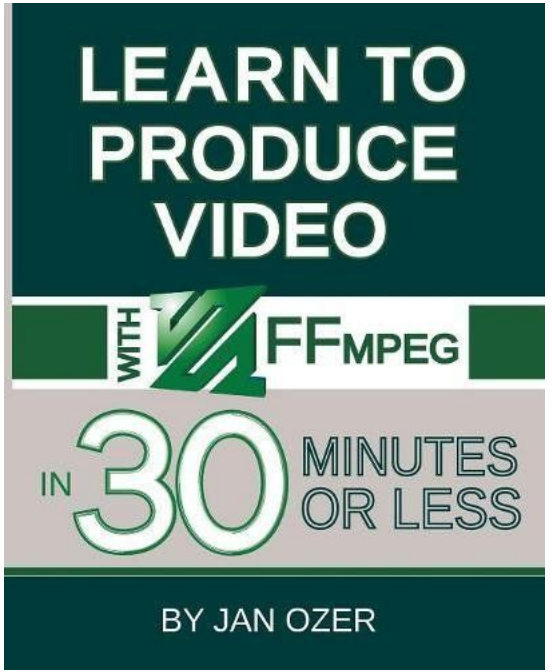
`-x265-params` – start of x265 commands, in x265 syntax

  - http://x265.readthedocs.io/en/default/
- One string of commands, separated by colon, no spaces until finished

- Preset, an (audio no), pass, format, and Null outside of this structure
- Scaling commands outside of –x265-params structure

FFmpeg Learning Resources

- Includes H.264/H.265
  - Creation of variant playlists with FFmpeg
  - Variant/master playlists with Apple tools
  - No Bento
  - No cloud stuff
- D103 - HOW TO: Building a More Robust Cloud Encoder With FFMPEG & More
  - Thus - 1:45 – 2:30



http://bit.ly/ffmpeg_30

Introduction to Bento4

- What it is: A fast, modern, open source C++ toolkit for all your MP4, HLS, and MPEG DASH media format needs
  - https://www.bento4.com/
  - Documentation for HLS - https://www.bento4.com/developers/hls/
- What you can do with Bento4
- Bento 4 vs. FFmpeg
- HLS options and Bento4 syntax

What can I do with Bento4?

- HLS generation, including master manifests, stream level manifests, mpeg-2 ts files, and fMP4 (fragmented MP4)
- MP4 to fMP4 conversion
- DASH generation
- Parsing and multiplexing of H.264 and AAC streams
- Support for DRM (Marlin, PlayReady, Widevine and FairPlay).
- Support for H.264, H.265, AAC, AC3, eAC3, DTS, ALAC, and other codec types.
- Dual generation of HLS and DASH from fragmented MP4
- Atom/box editing, and stream/codec information
- A lot more… https://www.bento4.com/

Bento4 vs FFMPEG

- Bento4 focuses on MP4 based content: Packaging & Transmuxing
- FFMPEG is a broad spectrum tool for media conversion, encoding & packaging

HLS options

- Master playlists
- Single file output with byte range requests
- I-Frame only playlists
- AES encryption
- DRM
- Audio stream sidecar
- Subtitle sidecar
- fMP4

Create Multiple Bitrate Assets

```
mp4hls --hls-version 4 input_7000kb.mp4 input_5000kb.mp4 input_3500kb.mp4
```

- Outputs:
- Master.m3u8
- Stream.m3u8 for each bitrate
- Iframe.m3u8 for each bitrate
- ts fragments for each bitrate

Multiple Audio Streams

`mp4hls video.mp4 spanish_audio.m4a` (different audio file)
`mp4hls video.mp4 [+language=es]audio.m4a` (multiplexed audio file, getting the spanish stream)

Outputs:

- Master.m3u8
- Stream.m3u8 for video and audio
- Iframe.m3u8 for video and audio
- ts fragments
- Audio.m3u8 and aac fragments

WebVTT Subtitles

```
mp4hls video.mp4 [+format=webvtt,+language=en]english.vtt
```

Outputs

- Master.m3u8
- Stream.m3u8
- Webvtt manifest and .vtt file

Encryption and Single Segment

```
mp4hls --hls-version 4 --output-single-file --segment-duration 6 --encryption-mode AES-128
  --encryption-key abaa09cd8c75abba54ac12dbcc65acd7 --encryption-url
  http://getmyKey?token=token video.mp4
```

Outputs

- All HLS assets (master, stream with byterange requests, iframe, single ts file)
- Assets are encrypted with AES-128, and encryption URL is added to the stream manifests
- Segment duration will be set to 6 seconds, but will only segment at the closest i-frame

Dual HLS and DASH From fMP4

```
mp4fragment input.mp4 output.mp4
```
(converts mp4 to fmp4)

```
mp4dash --force --hls --no-split --use-segment-timeline output.mp4
```
(without --no-split it will output .m4s segments)

Outputs

- Master.m3u8
- Audio.m3u8
- Video.m3u8
- Stream.mpd (DASH manifest)

Dual HLS and DASH From fMP4

DEMO

Let’s see this happen

Example Master Playlist for Single Bitrate

```
#EXTM3U
#EXT-X-VERSION:6
# Media Playlists
# Audio
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="audio/mp4a",LANGUAGE="en",NAME="English",AUTOSELECT=YES,DEFAU
   LT=YES,URI="audio-en-mp4a.m3u8"
# Video
#EXT-X-STREAM-INF:AUDIO="audio/mp4a",AVERAGE-BANDWIDTH=3454711,BANDWIDTH=4209761,CODECS="avc1.
   640020,mp4a.40.2",RESOLUTION=1280x720 video-avc1.m3u8
```

Other Info

- Bento will only segment at an i-frame
- Creates HLS assets faster than ffmpeg or shaka packager
- Gathers its metadata while segmenting, so codecs, average bandwidth, bandwidth, and resolution are automatically added to the manifests
- A full set of DASH and metadata options

List of all Bento4 binaries: https://www.bento4.com/

VOD: Server-based HEVC/HLS Asset Generation

- Overview
- Sizing your server
- Our experience
- Hardware starting point
- GPU pipeline
- Getting the software

Implementing Steps

- VOD: Server-based HEVC/HLS asset generation
- Cloud workflow
- Scaling
- Cloud encoding (the server)

OVERVIEW

- Choose your Cloud:
  - AWS
  - Azure
  - RackSpace
  - IBM SoftLayer
- Or don’t (On-prem)
- Or a hybrid (e.g. - On-prem and S3)

SIZING YOUR SERVER

- General
  - What general bitrates are you dealing with?
- Live
  - How many concurrent live streams?
  - Are you also transcoding optional renditions for ABR?
- VOD
  - How many concurrent videos being processed?
  - Is it transcoding or just transmuxing?
  - Do you need to create sidecar assets?

OUR EXPERIENCE

- In AWS we've found m3.large to be a pretty cost effective, decently performant and reliable instance size
- We made our decision in Azure based on AWS and went with as similar a match we could find, DS2_V2
- We use Linux as our base since it's friendlier with our software stack. Mostly RHEL.

STARTING POINT

- Get started with ec2 instances: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

- Get started with Azure VMs: https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-quick-create-portal/

GPU PIPELINE

- Offload processing from CPU to dedicated hardware
- FFmpeg has some support for GPU Acceleration
- You need to have specific supported hardware
  - Example: AWS EC2 g2.2xlarge + CUDA + FFmpeg with -hwaccel option specified

GETTING THE SOFTWARE

- You'll need to download and install software
- Our preferred toolset:
  - Bento4/FFmpeg (Video processing and Static Builds are easy install)
  - ImageMagick (spritesheets, thumbnails and image manipulation)
  - Node.js (You need an application server wrapper)
  - MongoDB (You need some data persistence)
  - Cloud Provider SDK (e.g. AWS SDK for JavaScript in Node.js)

Cloud Workflow: Making it Happen

- Designing a workflow API
- Workflow: file transfer
- Workflow: queue
- Open source libraries
- Sample code

DESIGNING A WORKFLOW - API

- You need a good workflow architecture
- Similar to AWS Simple Workflow Service for logical and atomic chunks:
  - Workflow (End to End Execution)
  - Steps (Ingestion, Processing, Transfer)
  - Tasks (Create alternate bitrate rendition, Thumbnails)
  - Adapters (We added this to be agnostic.
    E.g. AWS S3 vs. Azure Blob vs. On-prem)

WORKFLOW: FILE TRANSFER

- Try to leverage any performance enhancements available

- Day to Day Ingestion
  - AWS Multipart Upload
  - Azure Streaming Put a BlockBlob

- Initial Content Migration
  - AWS Import/Export Snowball
  - Azure Import/Export Service

WORKFLOW: QUEUE

- Gracefully handle all your users
- Processing takes time. You need to line up requests.
- Queuing w/persistence also lets you keep track of job status and what's pending in case of restart.

OPEN SOURCE LIBRARIES

- When there's a vibrant community you never have to reinvent the wheel

- We use Node.js which has node modules.
  - aws-sdk: AWS JavaScript Library for Node.js
  - fluent-ffmpeg: A node wrapper for the FFmpeg command line tool

SAMPLE CODE

- Check out the demo: https://github.com/realeyes-media/demo-encoder
- Here's a snippet

```
input.inputOptions = options.inputOptions;
output.outputOptions = ["-hls_time 8", "-hls_list_size 0", "-bsf:v h264_mp4toannexb", "-threads
0"];
input.inputURI = path.join(__dirname, '../../' + options.inputURI);
output.outputURI = `${directory}/${options.fileName +
options.timestamp}_${bitrate}.${options.outputType}`;
options.outputURI = output.outputURI;
output.outputOptions.push(`-b:v ${bitrate}k, `-r ${options.fps}`);

// Use options to call ffmpeg executions in parallel
executeFfmpeg(input, output)
```

Scaling

- Scaling and concurrency
- Scaling – multiple instances
- Multi-instance balancing
- Auto-scaling
- Container swarms

SCALING & CONCURRENCY

- How high can we go?
  FFmpeg will not error when the CPU is busy, just takes longer to process.

- First - Determine the Scenario:
  - The volume of files you need to simultaneously process
  - The average size of the files you need to process
  - The processing time that's acceptable for you org
  - The kinds of operations that need to occur (e.g. Just transmux? Transcode to 4 renditions?)

- Second - Run Performance Tests

SCALING - MULTIPLE INSTANCES

- Bigger instance or more instances?

- Bigger Instance
  - PRO: Handles more concurrency
  - CONS: Can be more costly
- More Instances
  - PRO: Cheaper - Can be scaled up and down to only pay when needed
  - CONS: More complicated to manage

MULTI INSTANCE BALANCING

- Scale Horizontally Transparently
  Clients hit a load balancer
- You can add more instances as needs grow in a transparent and simple way
- If your architecture is sound there's no need for session stickiness between the clients and the transcoding system
- AWS Elastic Load Balancer: https://aws.amazon.com/elasticloadbalancing/
- Azure Load Balancing: https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-load-balance/

AUTO-SCALING

- Leverage Auto Scaling Features
- Automate the spin up/down of instances based on a number of criteria:
  - Instance Load
  - Periodic Need for Faster Processing
  - Time of Day
  - Specific Events
- AWS Auto Scaling: https://aws.amazon.com/autoscaling
- Azure Auto Scale: https://azure.microsoft.com/en-us/documentation/articles/cloud-services-how-to-scale-portal/

CONTAINER SWARMS

- Docker is all the rage. Swarms and Service Discovery
- Create a swarm of Docker containers for a highly repeatable processing server snapshot that utilizes system resources efficiently
- Further increase automation through service discovery
- Implement “auto scaling” on steroids

Cloud Encoding (The Server)

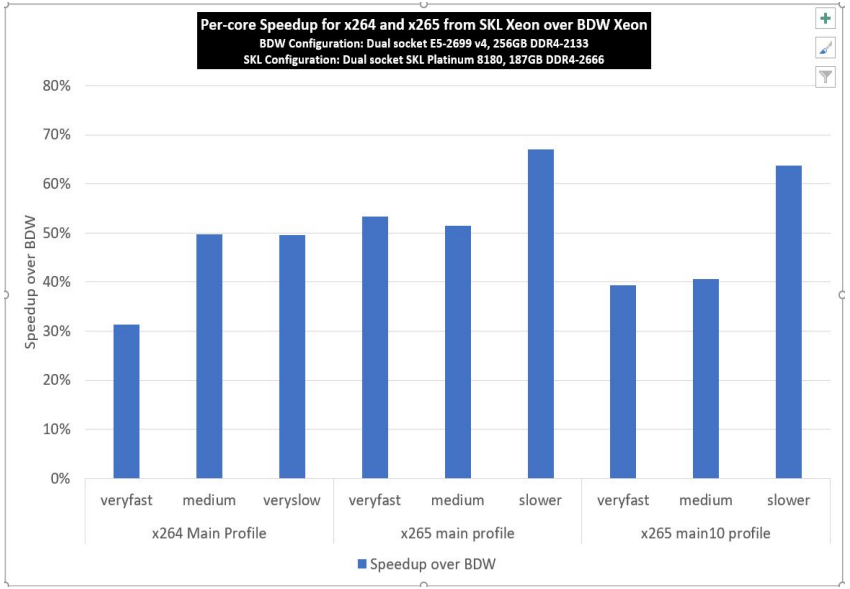
- >>> DEMO <<<

LIVE: Streaming with HEVC/HLS

- x265 Boost from Intel Xeon Scalable processor family
- Wowza
- Encoding – basically it comes down to hardware or cloud

HEVC Live – Intel Scalable Processor Family

- [x265 Boost from Intel Xeon Scalable Processor Family]
- x265 show a 67% average per-core gain for encoding using HEVC Main profile
- 50% average gain with Main10 profile across different presets



HEVC Live

- Wowza: https://www.wowza.com/docs/how-to-stream-using-hevc-h-265-transcoding

HEVC Live

- Live 4K HEVC/H.265 Software Encoding
- Haivision demoed live 4Kp60 HEVC software-only (x265) performance video streaming w/off the shelf hardware
- In the end it all comes down to hardware for live

More Demos

- Manifest Demo
- Playback demo and discussion (H.265 only)
- Playback demo and discussion (mixed H.264 and H.264)
- Playback demo and discussion (H.264 only)
- Additional resources

Manifest Demo: Walking through VOD and LIVE HEVC/HLS during playback (manifest viewer)

Manifest Demo: Walking through VOD and LIVE HEVC/HLS during playback (manifest viewer)

Playback Demo/Discussion: H.265 only

Playback Demo/Discussion: Mixed H.265 + H.264

Playback Demo/Discussion: H.264 only

Resources

- Slides: http://bit.ly/2gwlYs5

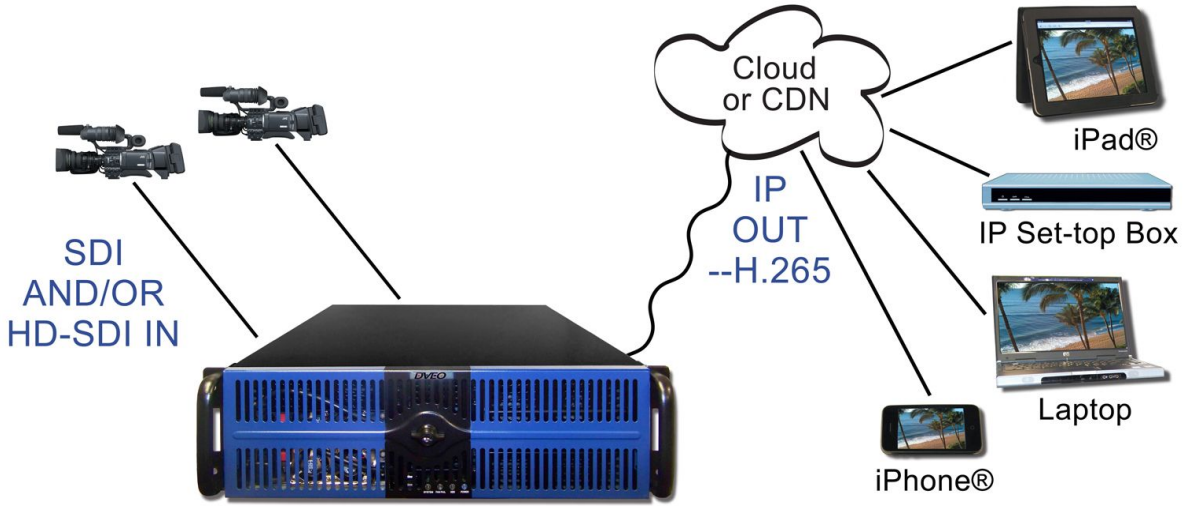
Third Party Alternatives

- Live
  - Full transcode and package
  - Contribution
  - Cloud transcode
- VOD
  - Appliance
  - Software
  - Cloud

Live: Full Transcode and Package

- DVEO Gearbox265
- Elemental Live
- Harmonic Electra XT
- Harmonic VOS Cloud Software
- Telestream Vantage Lightspeed

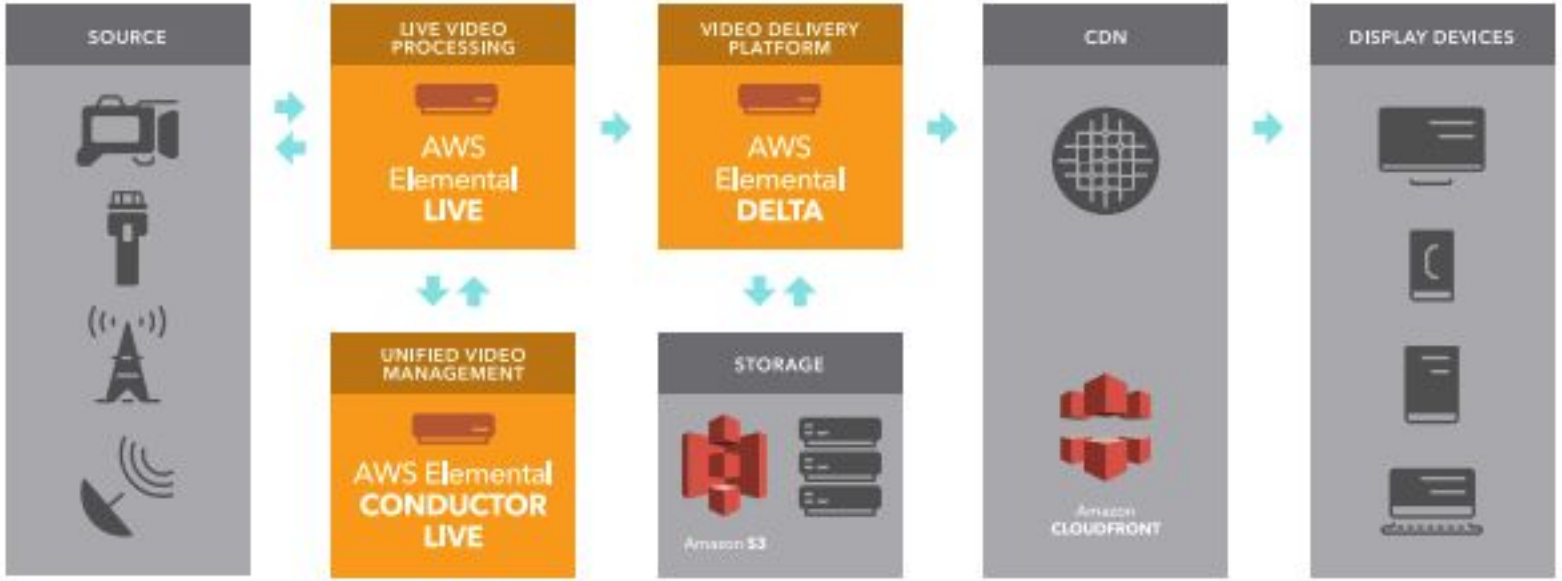
Full Transcode and Package: DVEO Gearbox265



- Hardware appliance

- No pricing info on website
- At Streaming Media West

Full Transcode and Package: Elemental Live



- Linux-based software; deploy anywhere

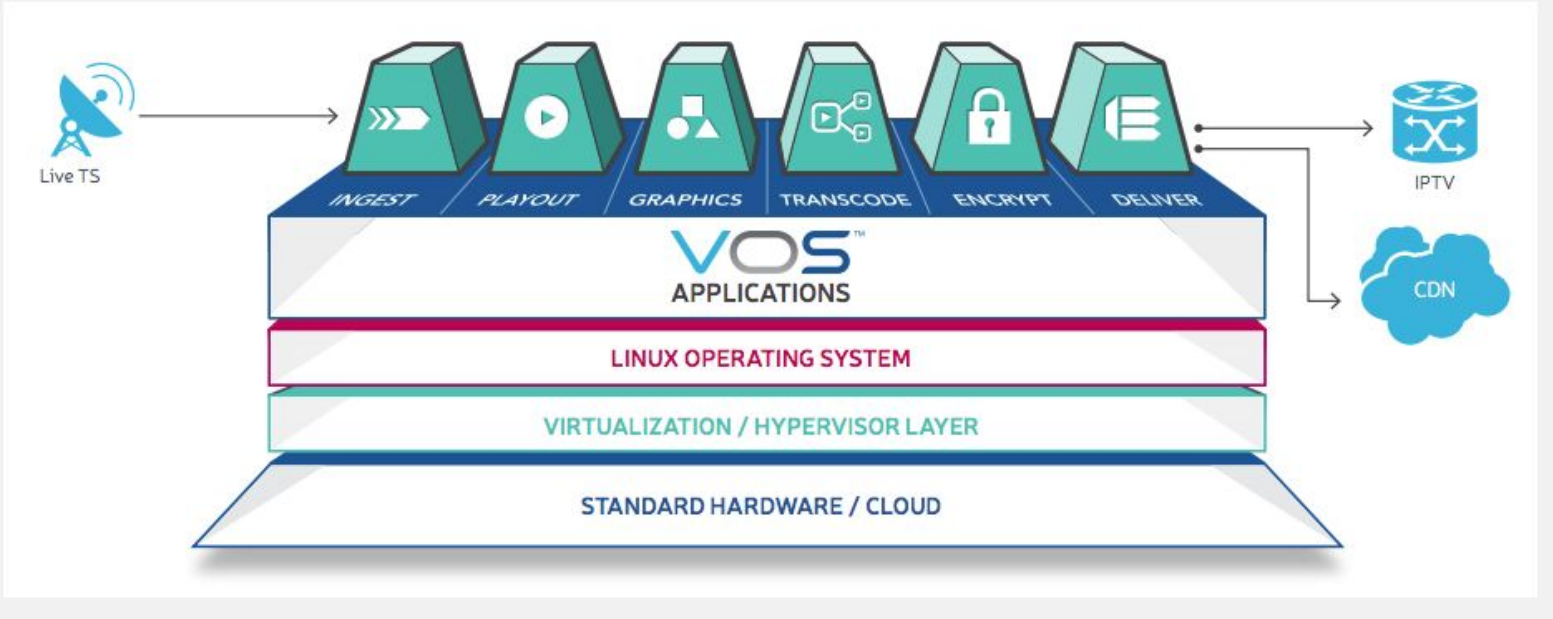
- No pricing info on website
- At Streaming Media West

Full Transcode and Package: Harmonic Electra XT, X2, X2S, VS

- Linux-based software; deploy anywhere

- No pricing info on website
- At Streaming Media West

Cloud Transcode: Harmonic VOS Cloud Software



- Licensed software
- Deploy in OpenStack or AWS

- No pricing info on website
- At Streaming Media West
- Live and VOD

Full Transcode and Package: Telestream Lightspeed Live Stream

- Linux-based software; deploy anywhere
- No pricing info on website
- At Streaming Media West

Live Contribution

- Harmonic
- LiveU
- Teradek

Cloud Transcode: Harmonic ViBE 4K

- Hardware/VOD
- Needs external packager for HLS
- No pricing info on website
- At Streaming Media Westn

Contribution: LiveU

HEVC Pro Card
(for LU) 600
$2,790
(Ethernet)

Cube 755
$2,990
(Ethernet +
Wi_Fi)

Slice 756
$3,990
(Ethernet +
Wi_Fi)

Contribution: Teradek

Cube 705
$2,790
(Ethernet)

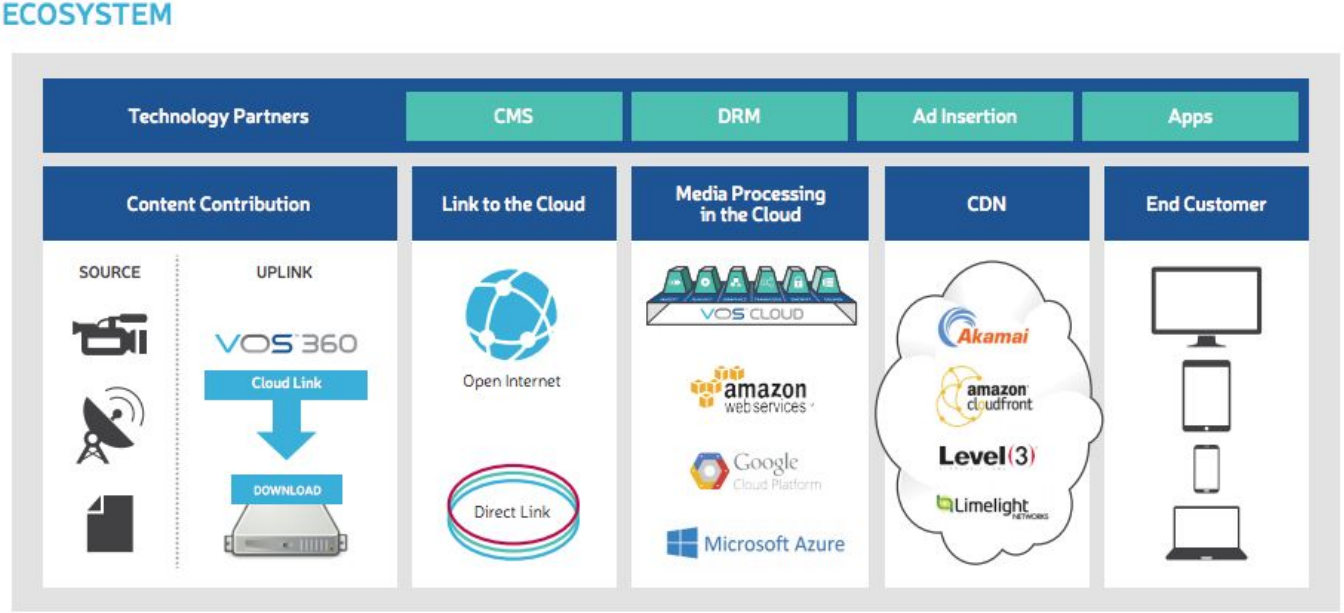
Cube 755
$2,990
(Ethernet +
Wi_Fi)

Slice 756
$3,990
(Ethernet +
Wi_Fi)

Live Cloud Transcode

- Harmonic VOS 360 cloud service
- Wowza

Cloud Transcode: Harmonic VOS 360 Service



- Linux-based software; deploy anywhere
- No pricing info on website
- At Streaming Media West

Wowza

- Can transcode to HEVC/not yet compliant with HLS spec
  - No CMAF yet
  - Here at show; ask when they will have

HEVC, HLS, and Live Production: A Wowza Interview

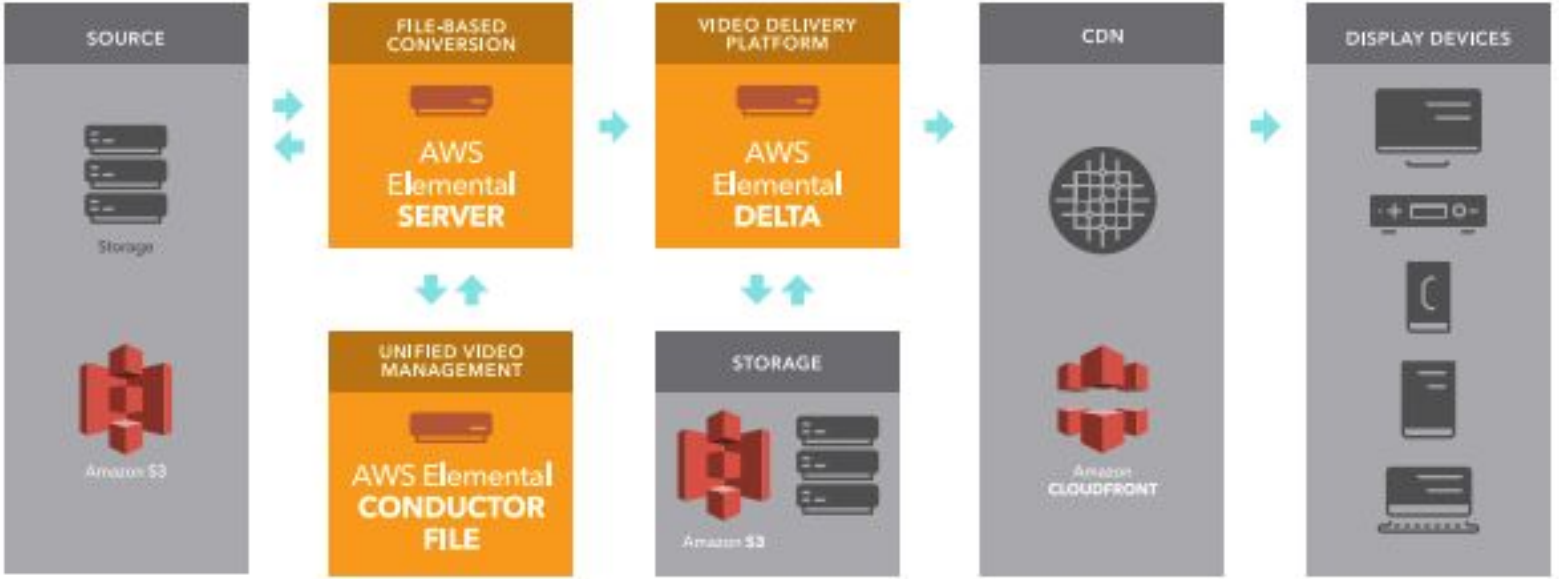
Wowza VP of Engineering Barry Owen

http://bit.ly/wz_hls

VOD

- Appliance
- Software
- Cloud

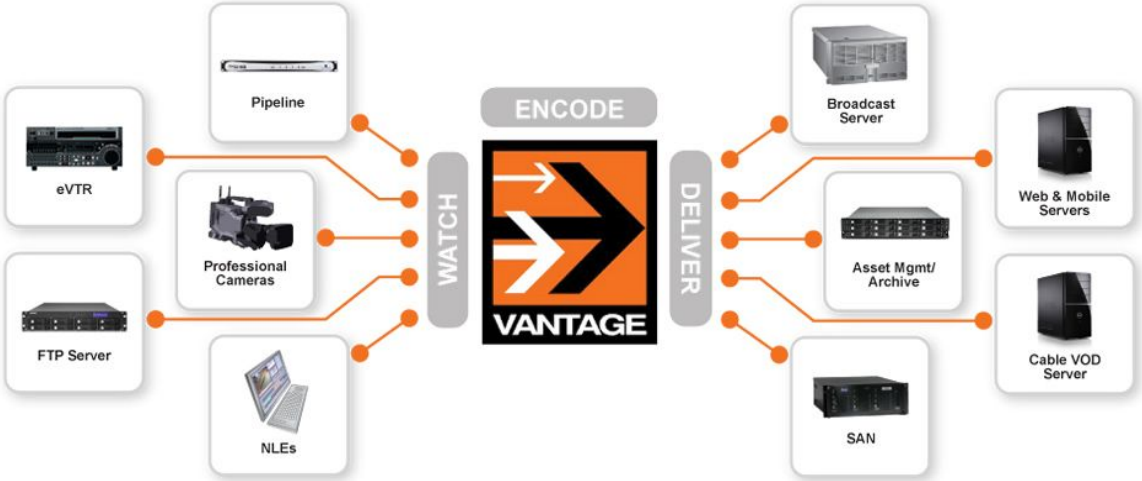
Appliance: AWS Elemental Server



- Linux-based software; deploy anywhere

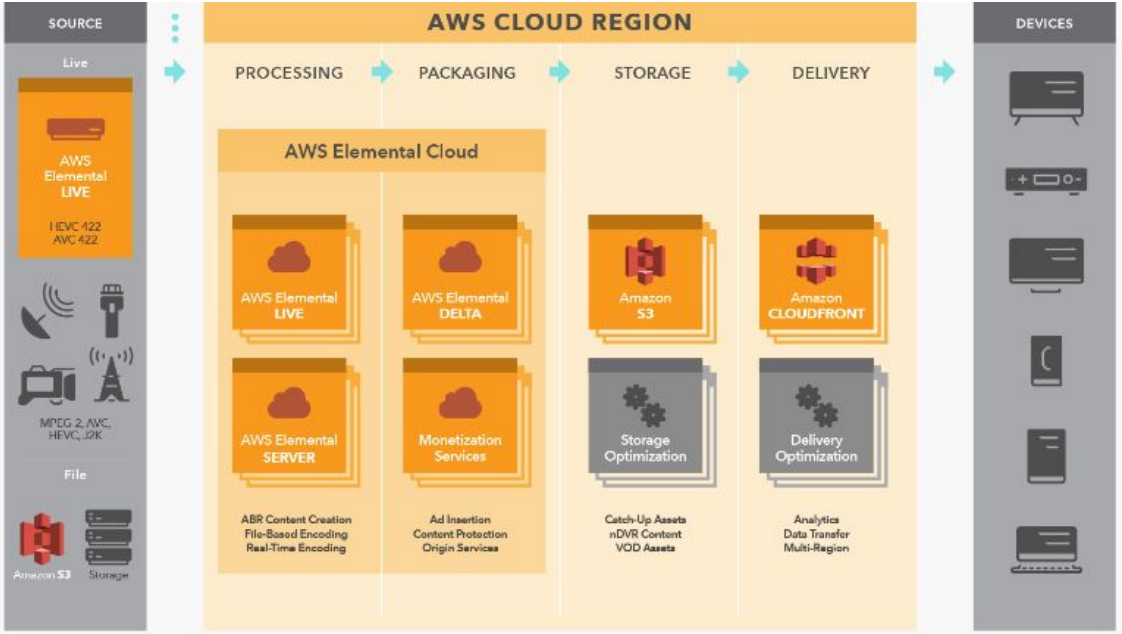
- No pricing info on website
- At Streaming Media West

Software: Vantage Media Processing Platform



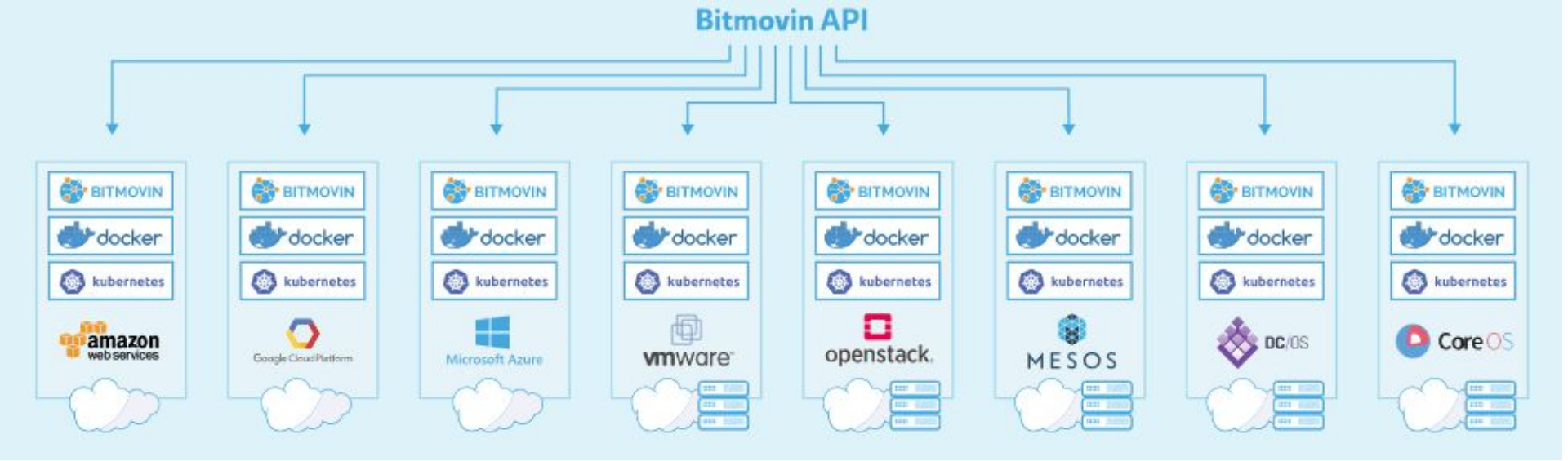
- Can run on servers or on public and private virtualized infrastructures
- At show

Cloud: AWS Elemental Cloud



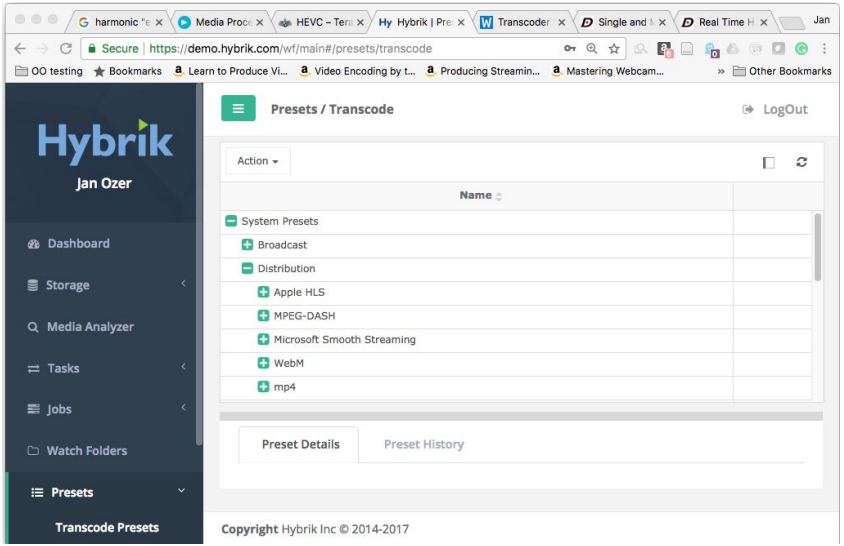
- True cloud-based product; extensible with other products
- No pricing info on website
- At Streaming Media West

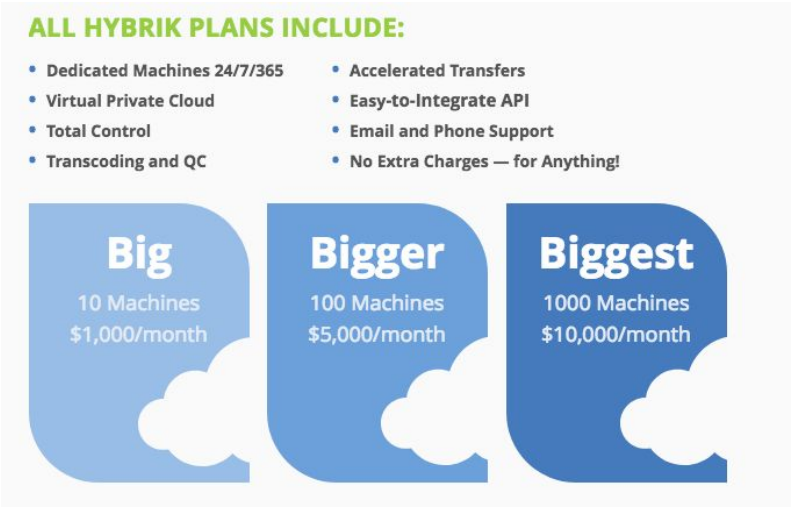
Software/Cloud: Bitmovin Video Encoding



- Available as a SaaS offering or for internal deployment
- No pricing info on website
- At Streaming Media West

Cloud: Hybrik Cloud





- Currently VOD; moving to live
- At Streaming Media West

Other Vendors

- Live
  - Contribution
    - Vitec – multiple encoders
- VOD
  - SDKs
    - Beamr
    - MainConcept
    - Multicoreware