Agenda

- Intro to key media concepts: frames, resolutions, framerates, bitrates, codecs
- Key concepts to work with in KVS: MKV, H264/H.265.
- HLS/DASH: Delivering media to Web and Mobile Applications.
- Multitrack media: video and audio, audio containerized media.
- Conclusion

Media Concepts:
Frame Rate, Resolution and Bandwidth

Frame Rate:
Streaming media is transmitted as a sequence of individual frames that describes the changing picture. The frame rate indicates the number of Frames Per Second (FPS) transmitted in a media stream.

Resolution:
The size (in pixels) of the transmitted frames. i.e.: 1080p (1920 wide x 1080 high).

Bandwidth:
Media bandwidth is measured in bits per second (bps) and is a simple calculation of **resolution (frame pixels) X frame rate x compression ratio.**

Media Concepts: Bits, Bytes and KVS Charges

KVS consumption cost is based on GBytes ingested by volume where as media bandwidth is a measure of network utilization that is measured in kbps / mbps or gbps. (bits per second).

There are 8 bits in a Byte and the capital ‘B’ in GB (opposed to Gb) indicates that the value is a multiple of 1024 (opposed to 1000).

Example:
KVS ingest volume on a typical 720p / 500 kbps media stream 24 Hrs per day:

- 500 kbps / 8 (to Kbytes p/s) / 1024 (to MBps) / 1024 (to GBps): 0.000059 GBps
- 0.000059 GBps x 60 (sec) x 60 (mins) x 24 (hrs) = 5.15 GB per day ($0.00850 GB).

KVS data ingestion cost (monthly): 1.33 USD

Media Concepts: Media Containerization

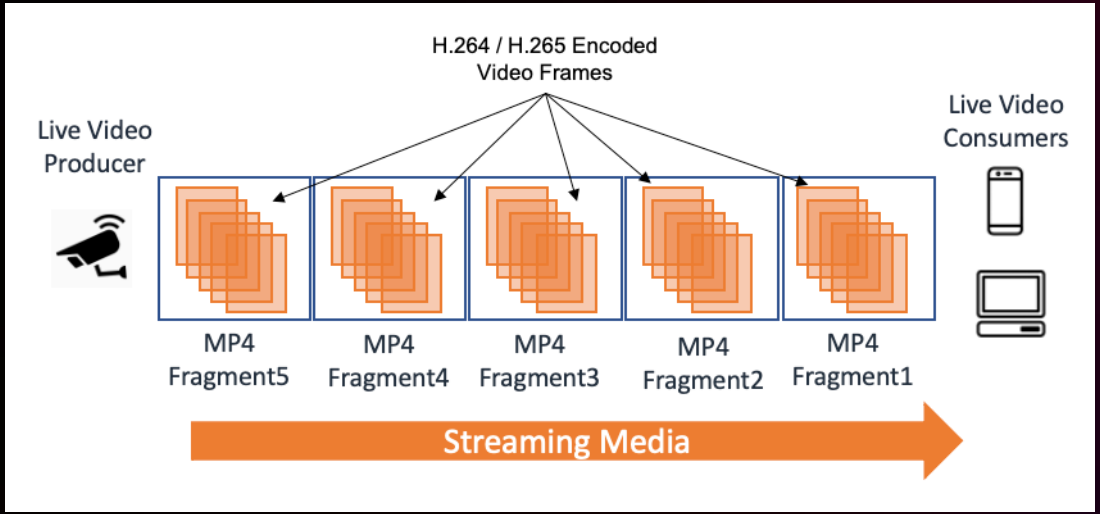
Media Containers:
For reliable and timely transmission of streaming media, groups of encoded video frames are containerized in one of a number of formats such as MPEG-4 (MP4), QuickTime Movie (MOV) or Matroska (MKV).

Media containers store meta-data like track numbers (video and audio tracks in the same container) , language packs, subtitle data and and closed caption files for the media stream.

A container may hold any kind of frame and encoding formats such as JPEG, H.264 and H.265.

Media Concepts: Video Fragmentation

Video Fragmentation:
Each containerized file is referred to as a fragment containing encoded frames and meta-data. This concept forms the basis of what we refer to as streaming media that allows users to stream live media or view file based media before the entire file transfer is complete.



Media Concepts: Video Formats and Encoding

Media Image Formats:
Digital media is presented as a series of changing pictures that can be in an uncompressed format such as X-Raw (think of a bitmap), a lossy compression format such as JPEG or lossless compression formats such as H.264 or H.265.

Lossless Vs Lossy Compression:
Lossless compression does not lose any data in the compression process and so the original image can be reproduced without any loss of quality.

Codec (Coder / Decoder):
Codec's contain the logic to preform compression and decompression of a digital signal (most often digital media). The Codec may be implement in software or dedicated hardware as is a computationally intensive process.

Media Concepts: Video Encoding / Format

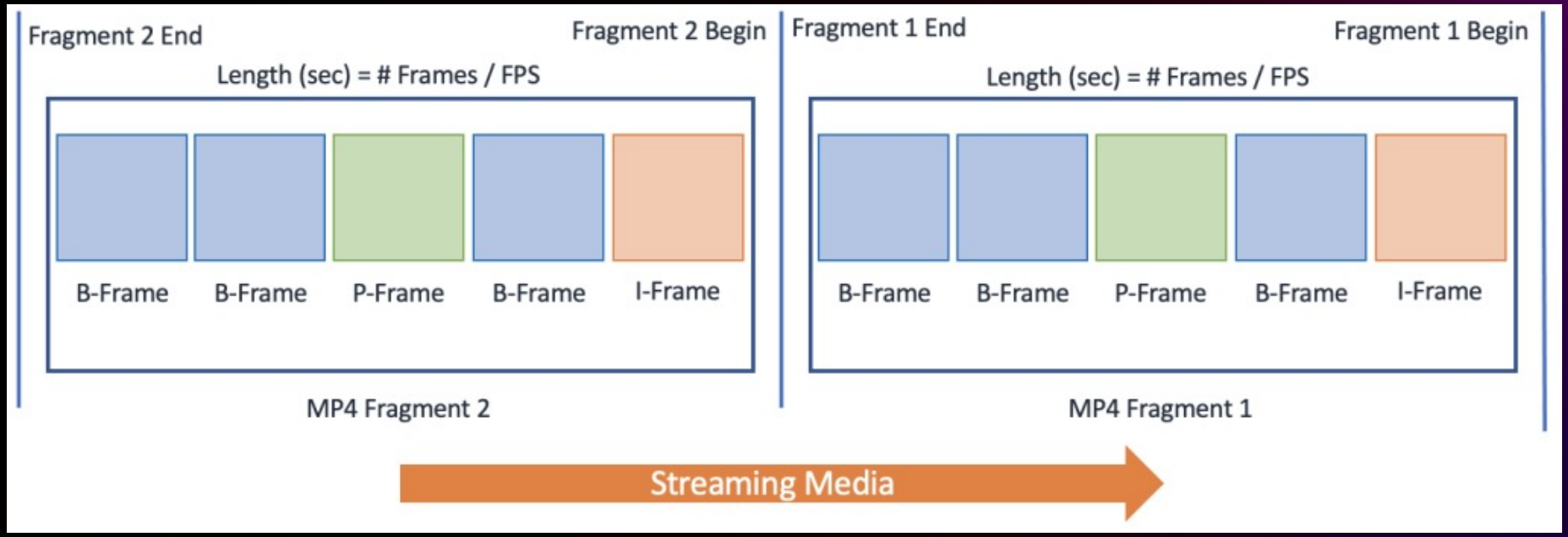
H.264 / H.265:
H.264 / H.265 video encoders achieve compression ratio as high as 2000:1 by only transmitting pixels that have changed between frames.

Each fragment must start with a key-frame (or I-Frame) which contains all data (pixels) and provides a reference for proceeding Predicted-Frames (P-Frames), and Bi-directional Frames (B-Frames).

It’s enough to understand that these frame types work together to describe the changing picture without transmitting unchanged data until the next key-frame is scheduled.

Media Concepts: Video Encoding / Format

Because the Key-Frame acts as a reference for proceeding frames types, a fragment is expected to start with a key-frame to ensure it is self-contained and not dependent on a proceeding fragment to play successfully.



Media Concepts: H.264 Vs H.265

H.264 Vs H.265 encoding:
H.265 is the proposed successor to the H.264 encoding format, it can increase compression ratios by over 50% but is also significantly more computationally intensive which also leads to higher power consumption.

However, H.264 can be used royalty free where as royalties are applied to H.265 use. Because of this, up until very recently; H.265 was poorly supported in many browsers and mobile based media players and so it is still not as commonly used.

KVS can ingest any time index data in MKV containers but to provide HLS / DASH replay (discussed later), the media must use H.264 or H.265 encoding for video and AAC encoding for audio.

Media Concepts: Transcoding

Where media encoding is processing uncompressed digital media into a compressed format, transcoding is processing from one compressed format to another.

A common example is to transcode H.264 to H.265. This is very computationally intensive and normally performed by either dedicated hardware or cloud services.

KVS does not preform transcoding. The media encoding that is ingested to KVS by the producer is exactly what will be provided to media consumers and so you must ensure end (user) devices support the encoding format being ingested.

Media Concepts: HLS / DASH

HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) are adaptive bitrate streaming protocols that expose streaming media to browsers, web and mobile applications.

Both HLS and DASH are supported by KVS and this is the simplest and most common means of providing end user media players an interface to KVS managed media. HLS is the most popular choice (an so we will focus on this here).

HTTP Live Streaming (HLS)
HLS divides media into streaming fragments which are made available to the requesting media player by a unique URL per fragment that is advertised via a manifest file generated by the HLS protocol itself.

Media Concepts: HLS / DASH

HLS Media Playlist (.m3u8):
This is a clear-text FIFO manifest file managed by the HLS protocol that contains media configuration tags and URL links to the live media fragments for media players to access. Below is an example extract of a HLS Media Playlist:

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-PLAYLIST-TYPE:EVENT
#EXT-X-TARGETDURATION:1
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="getMP4InitFragment.mp4?SessionToken==...&TrackNumber=1&SequenceNumber=1"
#EXTINF:0.498,
getMP4MediaFragment.mp4?FragmentNumber=91343852333181769835329731216345997453612345678&SessionToken=...&Tra
ckNumber=1&SequenceNumber=1"
#EXT-X-DISCONTINUITY
#EXTINF:0.5,
getMP4MediaFragment.mp4?FragmentNumber=91343852333181769840281491373487518686823456789&SessionToken=...&Tra
ckNumber=1&SequenceNumber=1”
```

Configuration / Initialization tags passed to the media player for various control and display functions.

URL / Links to FIFO list of available fragments

Media Concepts: HLS / DASH Containerization

HLS fragmented MP4 :
HLS / DASH accepts H.264/H.265 encoded frames stored in KVS and containerizes these using MP4 for transmission.

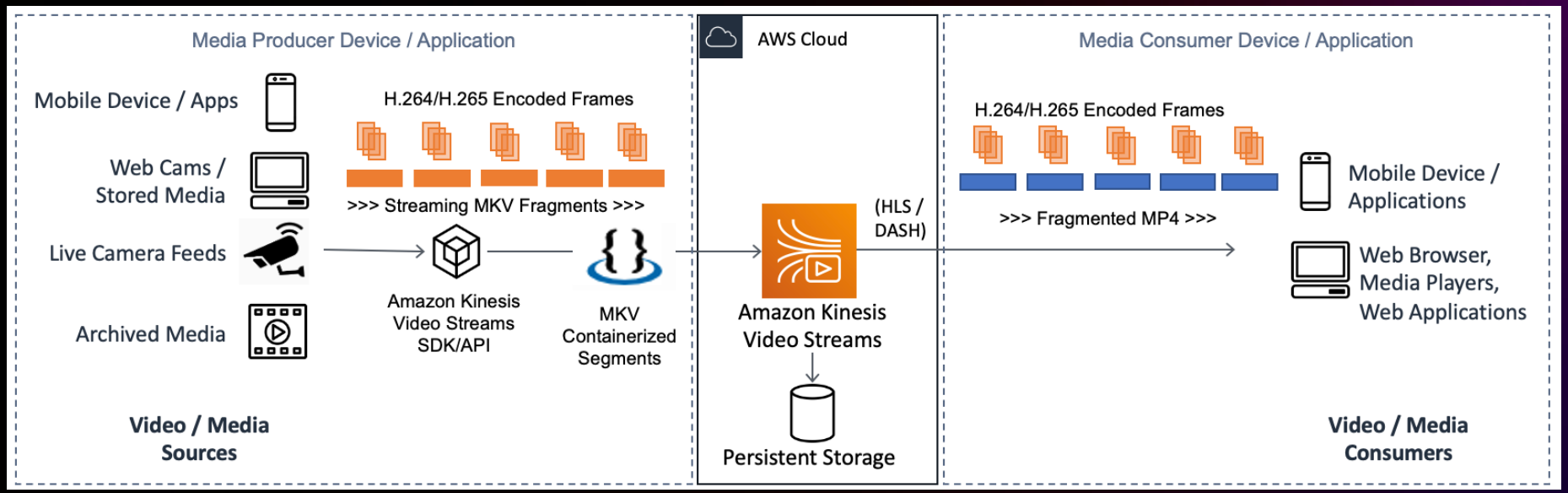
HLS Supported Encoding:
KVS can ingest and store any data formats however, to enable HLS/DASH sessions the ingested media must be H.264/H.265 encoded frames.

Consumer Media Attributes:
KVS does not perform any transcoding so HLS / DASH delivers the exact frames (images) that were ingested. This includes the same frame rate, resolution and fragment length and so the consumers media quality, bandwidth and latency is determined by the media ingestion source parameters.

KVS Common E2E Media Patterns

Media is ingested into KVS using the GStreamer KVSSink plugin, KVS Producer Libraries or the KVS API in Matroska (MKV) containerized fragments.

On the consumer side, live media is exposed to web and mobile applications using HTTP Live Streaming (HLS) delivered as fragmented MP4.



Multi-Track Media

Multimedia Container
A multimedia containers such as Matroska (MKV) used by KVS ingestion stores video, audio, metadata, synchronization and error correction tracks. That is, it can contain different tracks of various kinds of data but synchronize in a single multimedia flow.

Synchronised Video / Audio Tracks
Most commonly, multi-track media containers present two synchronised tracks, one for video and one for audio. KVS can ingest multi-track media in MKV containers. There are codec restrictions if HLS playback is required.

Time-indexed Data-store:
KVS is a fit for purpose data store for time index data and isn't limited to media formats. Multi-track media may contain synchronised audio, RADAR or LIDAR data.

Conclusion

Through this presentation we have covered the basics of media transmission and ingestion to KVS.

Groups of images are encoded and containerised for transmission as streaming media.

KVS supports ingestion of multi-track time indexed data in the Matroska container format. Typically this is video, audio or video and audio but can also contain RADAR, LiDAR or any other time indexed data.

KVS presents media to web and mobile applications using the well supported protocols of HLS or DASH. To present video as HLS, the encoding must be H.264 or H.265