Optimizing Video for Multi-Screen Delivery

Transcoding strategies for improved end-user experience, reduced network impact and lower overall costs

A NewBay White Paper
January 2011

newbay™

Disclaimer

The information provided in this Document is for information purposes only and shall not in any circumstances constitute or be deemed to constitute a warranty by NewBay as to the accuracy of such information, advice, statements or recommendations. No binding legal commitment shall be deemed to be in existence until such time as formal legal agreements are entered into by NewBay and NewBay shall have no liability whatsoever in relation to this Document and shall not be liable for any loss, expense, damage or claim arising out of the advice given or not given or statements made or omitted to be made in connection with this Document. All product names, company names and/or trademarks mentioned in this Document remain the property of their respective owners.

Table of Contents

Introduction

Consumers, now more technically aware and attached to their digital content than ever before, want to upload their media to secure, online storage and share it among their friends to be viewed on any device, anywhere and at anytime. Today, this is still "Internet domain" functionality, which doesn't have a ready-made answer to telco operator demands for predictable capital expenditure (CAPEX) and operating expenditure (OPEX), brand protection, high availability, a clear path to monetization and superior user experience.

Current solutions implement a basic form of content adaptation for video: pre-emptively transcode all audio-video content into multiple formats to suit a fixed range of device capabilities (codec implementation, device resolution, and video framerate) and available network bandwidth. Designed to overcome the latency associated with resource-hungry transcoding, this approach becomes prohibitively expensive when scaled for large networks; each individual upload requires 5 to 10 times of real storage space (compared to actual media size), which is not acceptable. Predictable CAPEX and OPEX are must-haves as operator services scale.

Running a video-centric service exposes the operator's brand reputation to new risks. Canonical transcoding solutions provide transcoding and nothing more, whereas the operator needs it to be fully integrated into a flexible ingestion chain, which will exclude such content as viruses/malware, inappropriate/illicit material, and copyrighted media, while supporting customized workflow tasks.

The operator needs a clear monetization path from its transcoding solution. The Internet-born phenomenon of "media snacking" presents opportunities for advertisement-funded business models framed around in-line and overlay advert delivery at the transcoding stage of ingestion. This may include in-stream cues which link to premium on-demand content, automatically selected by user demographic, behavior and location, or a simple operator logo overlay.

Above all else, the operator must adhere to the principle of "user experience is king." Past mistakes delivering mobile video (even via dedicated circuit-switched technologies such as 3G-324M) have shown the industry that end-users are largely intolerant of set-up latency, poor audio/video synchronization, and stuttering or blocky playback. Addressing such demands across a finite wireless spectrum is certainly demanding and requires a solution with intimate knowledge of media encoding, device capabilities and efficient delivery.

Content Adaptation Overview

Content adaptation is the process of modifying content from one representation to another. In the context of content adaptation, the transcoding of video content is a special case due to the large variety of output possibilities, some of which are discussed here.

Video is not a new technology. Analog and digital representations of moving images were originally designed solely for representation on forgiving television screens. These representations have had to adapt over the years to track the growing array of video-capable devices, advancements in device output capabilities and processing power, viewer expectations of output quality, and demands to squeeze more and more video content into distribution pipelines.

The basic principles of video representation have not changed – play a sequence of images in quick succession such that the viewer perceives the result as a smooth flow of motion. Simple. However, a massive amount of data is required to represent this image flow, which must be efficiently encoded for later decoding by the player, using a mechanism defined by a video codec. Many variations exist today, but the most popular encoding techniques include some of these steps:

- Apply a **color space,** which prioritizes according to human perception capabilities
- Remove **temporal redundancy** by finding similarities between neighboring frames
- Remove **spatial redundancy** by exploiting similarities within each frame
- Remove **statistical redundancy** by applying entropy encoding

These steps result in an encoded video stream with the following attributes:

- **Resolution** – the dimension of each frame in pixels
- **Framerate** – the number of frames to be played per second
- **Bitrate** - amount of data output by the encoder per second (variable or constant)

Thus, digital video representation has many degrees of freedom. This is compounded by the fact that the perceived quality of a video stream at the output device is entirely subjective. Transcoding of video for delivery to a range of output devices must address all of these, by combining:

- **Pass-through transcoding** – converting between codecs without modification
- **Trans-sizing** – converting from one resolution to another
- **Trans-rating** – converting from one bitrate or framerate to another

The level of freedom across encoding options and the wide range of network performance and device capabilities represent the core challenges when managing video content.

Industry Best Practices

Transcoding for Today's Multi-Screen World

"Multi-screen playback" is an industry term describing the transparent delivery of video of the appropriate quality to the end-user device: mobile, PC, television, tablet, *etc*. It is something of a misnomer. Each class of device actually contains a huge range of device capabilities, including: physical size, screen resolution, usable resolution, audio and video codecs implemented, transports supported, displayable framerate, input/output (I/O) bandwidth, *etc.* As an example, consider the wide range of resolutions and platforms (each with different levels of support for various codecs and bitrates) of a small selection of devices in Table 1. How should each piece of media be prepared and delivered to these devices?

Device	Resolution	Platform
Samsung Alias	176x220	BREW
Apple iPhone 3GS	320x480	iOS
BlackBerry Storm2 9550	360x480	BlackBerry
Google/HTC Nexus One	480x800	Android
Motorola Droid A855	480x854	Android
Apple iPad	1024x768	iOS

Table 1. Resolutions and Platforms for a Selection of Devices

Many solutions today implement a "lowest common denominator" solution to address this wide range of functionality – choose one content profile that works across all devices. Unfortunately, this impairs the end-user experience by forcing many of them to experience content in a format far below their expectations for their specific device.

Instead, content requested by a user must, as far as is practicable, be tailored to be viewed for the specific device in use. In general, this requires three steps:

1. Selection of an output resolution that is a close match to the physical resolution of the playback device, without distorting the aspect ratio of the original content.
2. Selection of the best available codec for that device. This may be based on codec complexity, but must consider the quality of the on-device player implementation.
3. Ensure appropriate framerate and bitrate. Mobile devices, in particular, have resource constraints that may determine if a stream can render correctly in the player.

Defining a reliable and objective measurement of video playback quality is elusive, particularly given that decoder implementations vary across devices and players. Therefore, a mathematical formula cannot be applied to the selection of transcoding parameters. The video delivery platform must know as much as possible about the nuances of the clients for which it will transcode and make appropriate decisions on what to deliver.

Transcoding as a Component of Content Ingestion

Although it is often the "heavyweight" in a multimedia processing platform, video transcoding is just one task of an ingestion chain of tasks that must be performed on uploads to a content cloud service. Such a chain might be performed as shown in Figure 1, with the chain managed by a workflow and task service (WTS) that orchestrates task execution while optimizing data flow and parallelizing tasks when appropriate. Important considerations for the flow of ingestion are listed below.

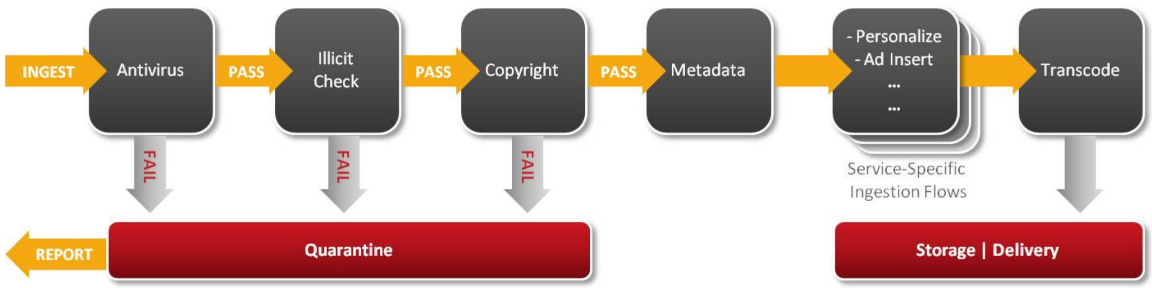

Figure 1: General Flow of a Sample Content Ingestion Chain

Antivirus Scanning

Viruses and malware are usually associated with executable content, not media. But, any ingested file might be an executable containing a virus, provided with a new extension or with an innocent MIME type. Alternatively, known player or transcoder vulnerabilities may be exploited by uploaded content to run embedded executable binaries that compromise the security of a platform deployment. A media-aware antivirus scanner is an essential component of all content ingestion.

Inappropriate Content

Inappropriate content is that which violates operator policy and/or local legal statutes: depiction of gratuitous violence, sexually explicit imagery or other policy violations. As with antivirus material, ingestion of such content must be automatically detected and refused at point of entry – or at least quarantined for manual checking.

Copyright

Copyright-infringing 'rips' of Blu-ray or DVD movies are likely to be a source of material to be ingested to any video storage/sharing platform and must be refused at point of entry. It is not sufficient to perform metadata-based digital rights management (DRM) checks, which may be easily worked around. A solution which checks a "digital fingerprint" of the media must be compared to a live database of known copyrighted material.

Metadata Manipulation

When available, metadata contained in media file containers offers useful information on the content producer, media title, album, genre, *etc.* An intelligent ingestion engine will not only use this information for labeling, classification, tagging, and indexing, but also enrich the file's metadata with other information gathered during the ingestion chain.

Speech-to-Text

Automatic recognition of speaker-independent, free-flowing speech (as opposed to trained or pre-determined grammar-based phrases) in an audio signal is not yet an exact science. Computer capabilities are still far from a human's ability to extract syntax and semantics from our massive diversity of languages, dialects, accents, intonations, and irregular speaking patterns – especially for a signal with multiple speakers or irregular background noise/music.

However, it is certainly accurate enough to extract a useful textual representation of an audio signal. Applied to the audio component of an ingested video, the transcription can be stored with the file for uses such as closed captioning, index and search, content-relevant advertisement overlays, *etc.*

Advertisement Insertion and Personalization

Transcoding is a convenient point in the ingestion flow for monetization of media by performing overlays of operator logo or other advertisements, background audio jingle, *etc*. Alternatively, tagging the video stream with information such as the ID of the uploader or applying some optional video filter offers the possibility for operator-defined personalization services.

Transcoding for Efficient Storage

Just-In-Time Transcoding vs. Caching: Striking a Balance

Video transcoding is processor-intensive. Typically, video streams require decoding into a temporal sequence of frames before they can be manipulated and re-encoded as desired. Doing so is complex, and, depending on the video resolution, may take far longer than the video's duration to complete on a typical home PC.

The consequent latency has meant that many content-adaptation solutions today rely solely on "pre-emptive" offline transcoding – for each ingested piece of video content, perform 'N' transcodes to match the 'N' required output devices. A good plan in theory, as storage is relatively cheap. But not cheap enough when you have a growing system that requires 5-10 real bytes of storage for every video byte uploaded.

Fortunately, improvements in central processing unit (CPU) power, faster I/O, dedicated digital signal processing (DSP) and codec implementations have paved the way for improvements in the scalability and cost-effectiveness of just-in-time (JIT) transcoding solutions. Just-in-time transcoding solutions are specifically designed for transcoding speed without compromising video quality and can be used to deliver client-appropriate and network-appropriate video adaptations incrementally – i.e., before the transcoding operation completes.

Therefore, there are trade-offs between the storage impact of pre-emptive transcoding and the latency impacts of JIT transcoding. A well-designed transcoding solution will incorporate both principles, using a “hot cache” and “cold cache” mechanism to store transcoded videos as shown in Figure 2 and described here:

1) Ingested video content is stored unmodified into some reference storage.

2) The video is pre-emptively transcoded into ‘N’ pivot formats and stored in a cold cache. These pivot formats are manually pre-selected to:

a) match the most popular output devices and/or

b) function as a useful intermediate format for subsequent JIT transcoding

3) Later, when a client device requests a particular video, its capabilities are determined by the user-agent header or some parameter passed to the service. If a suitable version of the requested video pre-exists in the cache, then it is delivered directly from the cache. Depending on the service configuration, the item might be promoted from cold to hot cache at this point.

4) Alternatively, if there is no suitable video in the caches, an available JIT transcoder is used to transcode from pivot format and deliver to the client device. Usually the output will also be added to the hot cache in parallel.

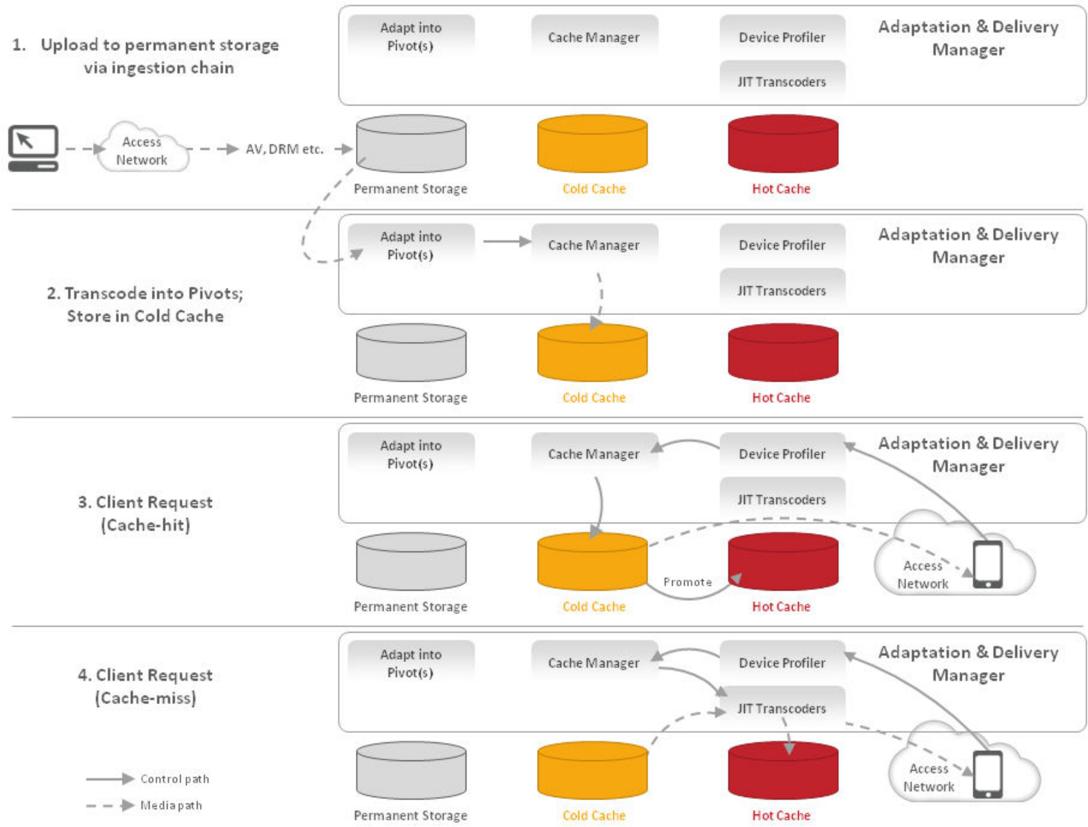

Figure 2: Overview of Video Adaptation Lifecycle

Large-scale content cloud services that implement an open sharing mechanism may opt to implement the hot cache, using a content delivery network (CDN) to offload network load to a geographically-distributed network of delivery servers.

A final consideration for JIT vs. pre-emptive video transcoding is the cost of the JIT video transcoding devices. They are not cheap, but the additional CAPEX pales in significance when compared to the cost of petabytes of potentially wasted permanent storage that can occur with a pre-emptive transcoding strategy, as presented in Figure 3.

Any financial comparison is somewhat dependent on assumptions and knowledge of the competing models. When comparing the CAPEX of JIT vs. pre-emptive video transcoding, the JIT model is shown to be more than 60 percent less costly than pre-emptive video transcoding. Both models presented here were based on the same active user and storage growth curves. The cost of the pre-emptive transcoding model was purely for transcoding storage overhead.

The cost of the JIT model included pivot storage plus the cost of the transcoding devices and on-demand transcoding (ODT) cache storage for pivots. The cost of the Adaptation & Delivery Manager (ADM) solution is not included in the overall calculation, but the impact is trivial when compared to the additional storage cost of the pre-emptive transcoding model.

The figures below show the difference in storage required for the two different models and the total CAPEX of each. As video scales, the cost savings only increase. Besides significantly reducing storage capacity requirements, ADM lightens the network load and associated costs and ultimately improves the user experience.

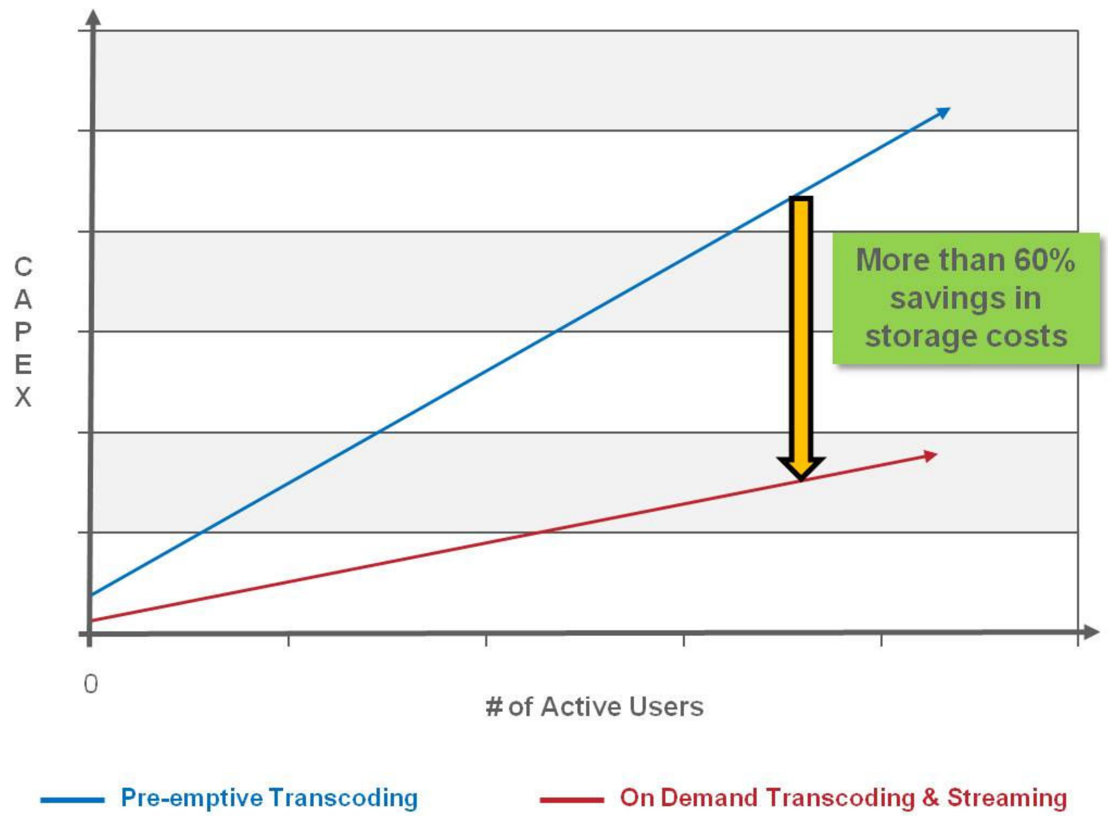

Figure 3: CAPEX Comparison: Pre-emptive Transcoding vs. JIT Transcoding

Socially-weighted Transcoding, Caching and Delivery

Earlier, we presented a sample video transcoding flow that provides an optimal balance between storage overhead (for pre-emptively transcoded videos) and delivery latency for a fixed number of JIT transcoders. Additional efficiency and intelligence may be layered over such a flow by integrating with the end-user's network of social networks.

Various social criteria may be used dynamically to determine suitable transcoding formats and to select storage for optimal delivery. Policy hints may be used like those presented as examples in Table 2 below for transcoding ingestion flows.

Social Criterion	Description	Hint provided
Device types of uploader's friends	The range of devices used by the uploader's friends	Most useful transcode targets
Uploader location	The geographic location most frequently occupied by the uploader	Populate these geo-local edge servers first
Friends' locations	Geographic location most frequently occupied by the uploader's friends	Populate these geo-local edge servers first
Uploader popularity	Uploader's number of friends	Likelihood of many hits; if high, prepare by transcoding as wide as possible
Uploader viral history	Uploader's history of uploading "interesting" or viral content	Likelihood of many hits; consider direct upload to CDN

Table 2. Social Criteria for Socially-weighted Transcode Caching

Transcoding and Delivery Mechanisms

Media Download vs. Streaming

There are many choices available for the delivery of audio/video media to the end-user, and careful consideration must be applied for any given deployment. In general, there are two classifications: **downloading** and **streaming**, but over time the boundary has blurred.

Download and Play

Downloading is the traditional approach to delivering media. The entire media file is transferred to the requestor, typically using a protocol such as hypertext transfer protocol (HTTP). Playback of the media cannot begin until the entire file is transferred – the main disadvantage.

Streaming

In contrast, **streaming** refers to the capability to transfer the content to the requestor in parts – typically in "near real time" – i.e., media is sent across the wire just before it needs to be rendered by the player. This means that playback may start almost immediately, although clients will wait briefly to buffer some media in case the connection degrades during playback. Popular streaming protocols include Adobe real time messaging protocol (RTMP) and IETF real time streaming protocol (RTSP) with media transfer by IETF real time protocol (RTP).

Aside from immediate playback, streaming protocols offer the advantage of random access – jumping forward and back in the media timeline. In addition, they integrate with network adaptation protocols such as RTP control protocol (RTCP), which allow delivery to be optimized in real-time based on bandwidth, packet loss, jitter, *etc*.

But streaming protocols are not without disadvantages. Because media is delivered in near real time, the control connection to the streaming server must be maintained for the full duration of playback. And, they typically encounter firewall issues. RTSP typically requires access to TCP port 554, and the media transport protocols need a range of open UDP ports, which firewalls do not typically provide. Consequently, a range of "middle ground" hybrid mechanisms have become adopted across the industry.

Hybrids

HTTP progressive download (HTTP-PD) was the first widely adopted solution to the latency pain of download-before-play. Assuming that all metadata necessary to play the media is contained in a header at the front of its container file, clients may start to play back the content before it is delivered entirely. This immediately improves user experience, but the user cannot jump forward in the media timeline beyond the media already delivered.

HTTP adaptive streaming (HTTP-AS) is a newer approach implemented by such standards as **HTTP live streaming** introduced by Apple™ and a candidate for IETF standardization via draft-pantos-http-live-streaming. It involves breaking up the original content into multiple chunks of short duration at transcode time, each into its own MPEG-TS container file. Thereafter, a standard HTTP server may be used to serve up a playlist pointing to these chunks and the chunks themselves. The playback client uses this playlist to download the correct chunks in turn, rendering a smooth continuous playback to the user. The main advantage is that the server may store multiple versions of each chunk at different bitrates, allowing the client to dynamically adapt to network conditions during playback. Microsoft™ **smooth streaming** adopts a similar approach.

Network Considerations

Network efficiency is another important consideration for operators when selecting video delivery mechanisms. Delivery of video over a lossy network will benefit from a streamed delivery mechanism which uses best-effort datagram transmission, avoiding the need for extra round-trips when packets are lost as per connection-oriented transfer mechanisms like HTTP download.

An un-throttled HTTP download of video will behave such that the client will consume as much bandwidth as it is allocated until the file transfer completes. In contrast, a streaming client will consume a lower bandwidth for a longer duration – typically the length of the media itself.

As an example, consider a video file (in some arbitrary container) of 10-minute duration, consisting of AAC-LC audio at 24Kbps, H.264 baseline profile video at 300Kbps and common intermediate format (CIF) resolution of 15 frames per second. The total file size is 24MB.

For the sake of illustration, Table 3 below indicates the difference in network behavior between a non-progressive download and streaming of the content over two different air interfaces. Note that streaming bandwidth is slightly higher than the media bitrate due to overhead associated with RTP headers.

Average Downlink Bandwidth	HTTP Download and Play		RTSP Streaming (at real-time)	
	Time before playback starts	Bandwidth usage	Time before playback starts	Bandwidth usage
400Kbps	8.2 minutes	400Kbps for 8.2mins	5 seconds	330Kbps for 10mins
1024Kbps	3.2 minutes	1024Kbps for 3.2mins	5 seconds	330Kbps for 10mins

Table 3. Comparison of Network Behavior for Download vs. Streaming

Content Processing with NewBay LifeCache

NewBay's LifeCache Platform exposes capabilities described in this white paper with the Adaptation and Delivery Manager (ADM) and Workflow and Task Service (WTS).

LifeCache Adaptation and Delivery Manager (ADM)

The ADM exposes RESTful HTTP interfaces for the adaptation of content, detection of client capabilities and delivery via client-smart, efficient, network-optimized mechanisms. The ADM may be used with a wide range of content as described for each of the content-specific modules shown in Table 4.

Audio and Video Module	Image Module	Document Module	Speech Module
Batch Transcoding	Thumbnail	Doc <-> PDF/HTML conversion	Convert Speech to Text
On-Demand Transcoding	Rotate/Crop	XLS <-> PDF/HTML conversion	Convert Text to Speech
HTTP/HTTP-Progressive Delivery	Special Effects (Red-Eye-Reduce *etc.*)		
RTSP/RTMP/iPhone Streaming	Convert Format		

Table 4. Modules and Features of the LifeCache Adaptation and Delivery Manager

The range of container formats and codecs supported by the ADM audio and video modules includes those listed in Table 5. Note that a small number of the codecs listed below incorporate behaviors, which may be patent protected and, as such, may require payment of royalties to the relevant constituencies for revenue-generating usage. However, this is highly dependent on the exact nature of the final service deployed and, where necessary, can be avoided by making the best use of free and open formats.

Category	Details
Container Formats	3GP, AVI, DIVX, FLV, MP3, MP4, MPEG2-TS, MPG, OGG, WAV, MOV, ASF
Audio Codecs	AAC-LC, AAC-HE, AMR-NB, AMR-WB, PCM, MP3 (MPEG-1, MPEG-2, MPEG-2.5 Layer 3), Vorbis, WMA
Video Codecs	H.263, Sorenson H.263, H.264 (MPEG-4 Part 10 AVC), MPEG-4 Part 2 Visual, VP6, WMV, Theora, XVid, MPEG-1, Indeo

Table 5. Selection of Supported Audio/Video Codecs and Container Formats

As an example use-case, consider a cloud backup service based on LifeCache Digital Vault (DV). A typical user might back up the "My Documents" or “Home” folder from his/her PC at home, which contains 5,000 music files (in MP3 and M4A format), 3,000 image files (in JPG format) and 20 video files (mixture of 3GP files from his phone, AVI files from his video camera).

Once these files are uploaded and stored securely, the user expects to subsequently browse through and render his content from any device – some of which may not be capable of the formats mentioned. Digital Vault uses the ADM service to make the relevant adaptations, store the output, and then later make a smart decision about what and how to deliver to the user's iPhone, TV, Xbox 360, Android handset, tablet and/or some other connected device.

LifeCache Workflow and Task Service (WTS)

The LifeCache Workflow and Task Service (WTS) efficiently orchestrates the execution of arbitrary tasks on content in the cloud and offers a series of "pre-rolled" modules, which implement behaviors commonly required for a cloud content solution. These modules are described in Table 6.

Anti-Virus Module	Copyright Module	Illicit Content Module	Search Module	Adaptation Module
Scan for Virus or other Malware	Perform Copyright Checks	Perform Illicit Content Checks	Index for Search	Pass to ADM

Table 6. Modules and Features of the LifeCache Workflow and Task Service

A common use-case for the WTS is during the ingestion of content into a cloud content service. Usually operators will require all of the tasks above to be performed on each file before the upload is fully complete. The modules described above can easily be chained together to implement the flow as shown in Figure 4. The WTS typically performs content adaptation at the end of the chain by calling out to the ADM.

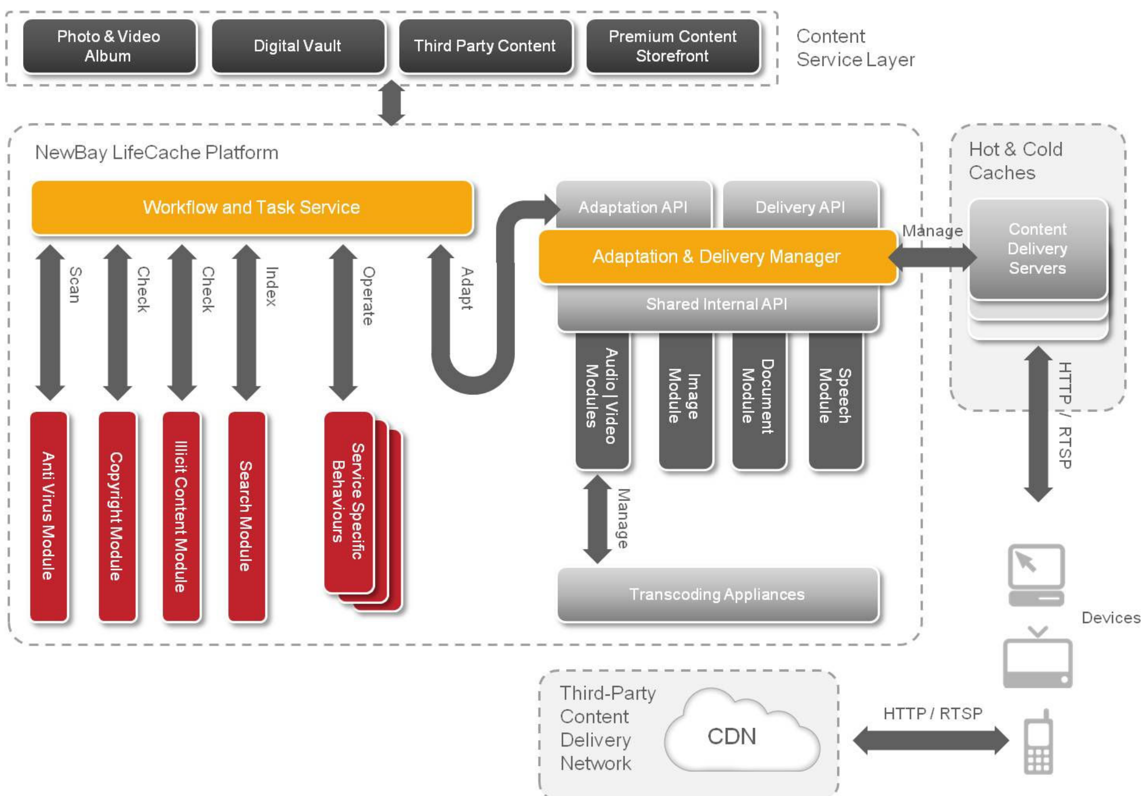

Figure 4: LifeCache Platform with ADM & WTS for Video Networks

ADM in IMS, LTE and RCS networks

The 3GPP IP Multimedia Subsystem (IMS) – whether or not it will be deployed in its entirety as originally intended – offers a valuable framework for the dissemination of media to end-users.

All of the transcoding principles discussed here still apply in an IMS network. Depending on the service provided, the ADM interconnects with the session initiation protocol (SIP)-based IMS core network. One simple approach is shown in Figure 5. Calls from mobile IMS clients place SIP calls to the core network, passing through various proxies before arriving at the application server defining the service. When media is required by the service, the SIP application server engages the media resource function (MRF) and requests playback of media using some SIP-borne control channel: NetAnn, Video-in-VoiceXML or the MediaCTRL media control framework from the IETF. In turn, the MRF pulls media from the Adaptation and Delivery Manager, which may stream the necessary media directly to the client.

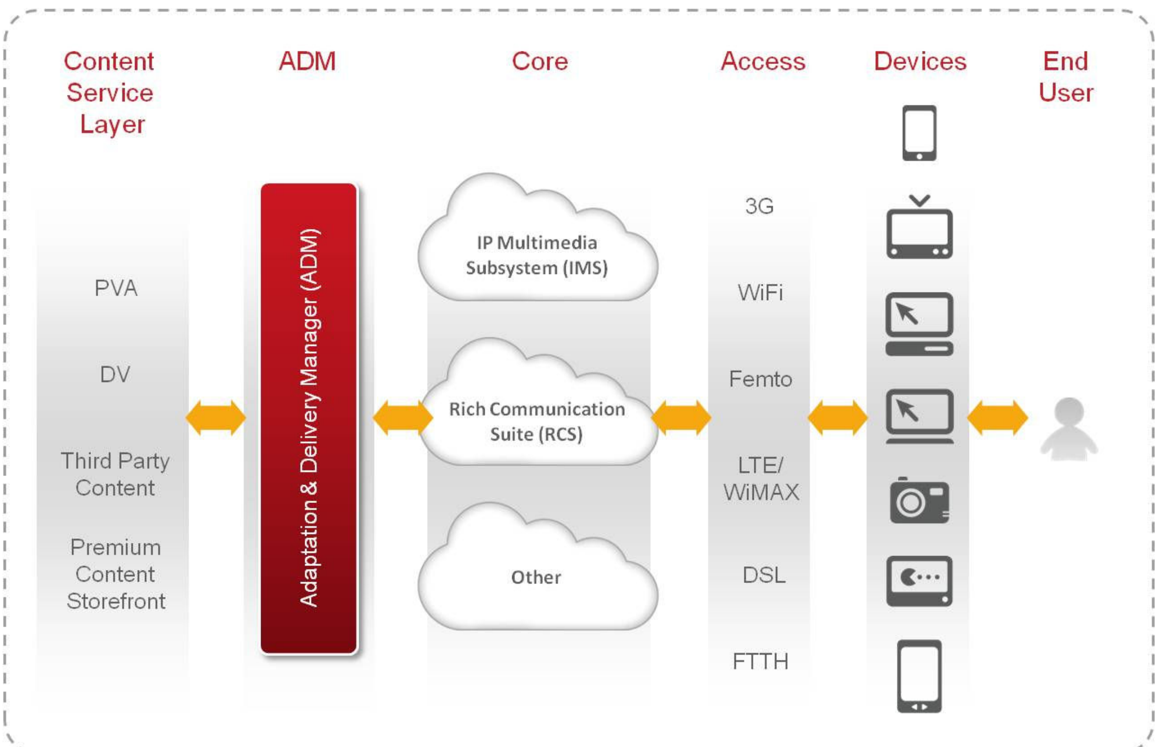

Figure 5: Leveraging LifeCache ADM for IMS, RCS and LTE Networks

The video ADM framework applies regardless of access network – LTE, 3G, WiMAX, cable, *etc*. Additionally, ADM can be leveraged wherever video is being uploaded by or delivered to an end-user, including premium content services (storefronts), GSMA's Rich Communication Suite (RCS) model, video streaming businesses, *etc*.

Summary

Consumers and businesses are devouring an ever increasing amount of content, especially bandwidth heavy video. Current video solutions implement a basic form of content adaptation – pre-emptively transcode all audio-video content into multiple formats to suit various device capabilities and available network bandwidth. This approach becomes prohibitively expensive when scaled for large networks. Fortunately, technological enhancements have paved the way for improvements in the scalability and cost-effectiveness of JIT transcoding solutions. When comparing the CAPEX of JIT vs. pre-emptive video transcoding, the JIT model is shown to be more than 60 percent less costly than pre-emptive video transcoding. As video scales, the cost savings only increase.

Video transcoding is just one task of an ingestion chain of tasks that need to be performed on uploads to a content cloud service. A flexible ingestion chain should exclude content such as viruses/malware, inappropriate/illicit material, and copyrighted media, while supporting customized workflow tasks (e.g., personalization, ad insertion, *etc*.).

The principles and capabilities discussed in this paper apply to various content types – user generated and premium – and various content delivery sources – online photo sites, digital vaults, third-party content services and premium content storefronts. Additionally, supporting systems and processes can be deployed as standalone solutions or as a supporting component in IMS, RCS and/or other content delivery services.

NewBay's LifeCache Platform exposes capabilities described in this white paper with the Adaptation and Delivery Manager and Workflow and Task Service. ADM adapts content to meet device and client capabilities and exposes smart, efficient, network-optimized delivery mechanisms to clients. Content ingestion is handled by the WTS, which makes use of the ADM if it needs to perform adaptation. WTS orchestrates task execution, while optimizing data flow and parallelizing tasks when appropriate.

In summary, operators, device makers and other video providers can deploy NewBay's LifeCache ADM and WTS solutions to improve the user experience, lighten the network load, significantly reduce storage capacity requirements and lower total cost of ownership.

About NewBay

NewBay is a leader in cloud-based digital content services, enabling subscribers to create, store, manage, view and share user content. NewBay LifeCache product suite (See Figure 6) empowers operators to deliver an integrated set of converged rich-media services across any Internet connected device such as mobile, PC, tablet and TV. NewBay's products include social networking, photo and video albums, digital vault, handset and desktop clients, smart address book, push notifications and messaging services. NewBay's products are built on the LifeCache Platform to uniquely provide telco-grade, scalable solutions.

NewBay enables operators to increase ARPU, drive messaging and data traffic, strengthen customer loyalty and build communities and social networks based on user content. NewBay is delivering profitable, highly successful commercial services for operators. Customers include: Telefónica O2, T-Mobile, France Telecom/Orange, U.S. Cellular, AT&T, Telstra, Verizon, Alltel Wireless, and LG Electronics. NewBay LifeCache is processing millions of messages daily and stores billions of media for live operator services. NewBay is based in Dublin, Ireland, with offices in Seattle, Palo Alto and Raleigh, USA; London, UK and Dusseldorf, Germany. NewBay was founded in 2002 and is privately held. Investors include Balderton Capital and Fidelity Growth Partners.

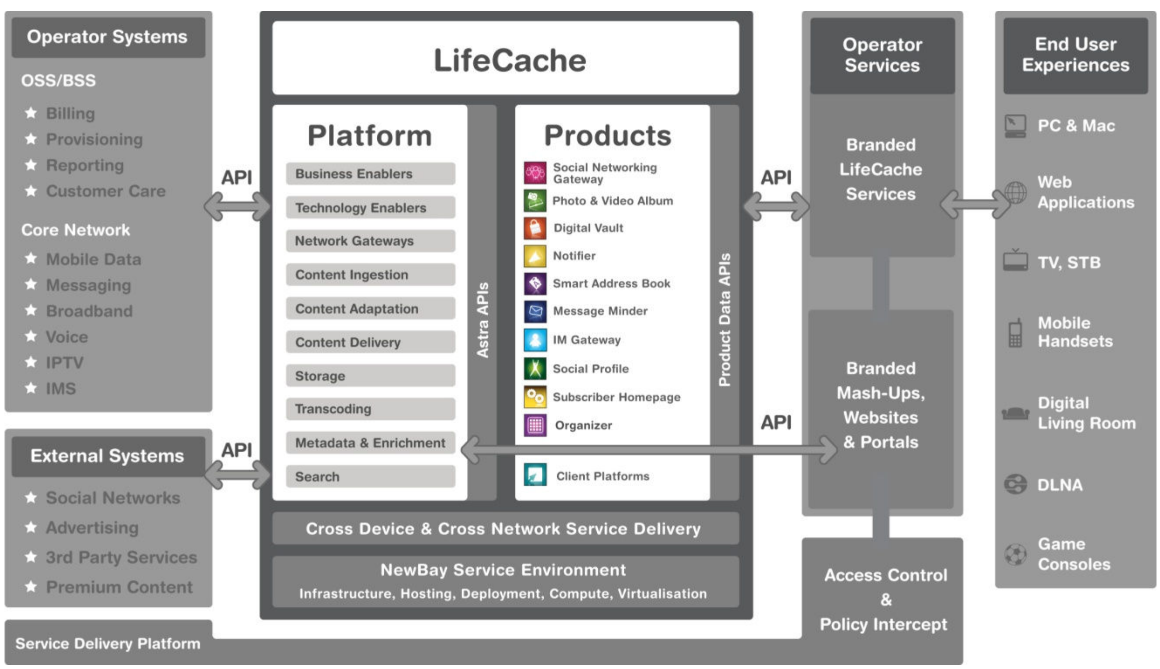

Figure 6: NewBay LifeCache Architecture

Appendix

Acronyms

AAC-LC - Advanced Audio Coding Low Complexity

ADM – Adaptation and Delivery Manager

CAPEX – Capital Expenditure

CIF – Common Intermediate Format

CPU – Central Processing Unit

DRM - Digital Rights Management

DSP – Digital Signal Processing

HTTP – Hyper Text Transfer Protocol

HTTP-AS – HTTP Adaptive Streaming

HTTP-PD – HTTP Progressive Download

IETF – Internet Engineering Task Force

IMS – IP Multimedia Subsystem

I/O – Input/Output

JIT – Just-in-Time

MPEG-TS - MPEG Transport Stream

MRF – Media Resource Function

OPEX – Operating Expenditure

RCS – Rich Communication Suite

RTCP – RTP Control Protocol

RTMP – Real Time Messaging Protocol

RTP – Real Time Protocol

RTSP - Real Time Streaming Protocol

SIP – Session Initiation Protocol

WTS – Workflow and Task Service

info@newbay.com
www.newbay.com
www.newbay.com/blog
www.twitter.com/NewBaySoftware
www.facebook.com/NewBay

