# Cloud Computing for Cable TV: Expanding Choice, Control & Content for a New Generation of Television Viewer

*John W. Callahan, Chief Technology Officer, ActiveVideo Networks*

*Jeremy Edmonds, Director, Product Management, ActiveVideo Networks*

## INTRODUCTION:

The multichannel television industry is caught between two significant realities: Growing consumer demand for increased choice and control of TV content, and an installed base of technologically fragmented, resource-limited digital set-tops and CE devices.

Consumers have become acclimated to the media richness, immediacy and dynamic nature of Internet video. It follows that consumers – particularly those under 30 years of age -- are increasingly seeking to avoid the usability limitations associated with traditional broadcast multichannel video delivery systems.

As a consequence, content owners, service providers, and advertisers alike face major challenges to traditional business models as video consumers shift to the one-to-one world of unicast, on-demand video streams that characterize VOD, DVR, and Internet video.

The notion of "just plug a broadband connection into your HDTV" – a predominant trend at the January 2009 Consumer Electronics Show -- is the latest step in the process of moving from the traditional operator business model of bundling access and content, to a world where the consumer chooses "what, when, and where" to "watch television" [as our parents called it]. The implications for cable are clear. First, this movement damages the operator's traditional revenue streams, i.e., bundling, "tiering" and 30-second ad insertion.

And second, the inexorable growth in IP access network bandwidth, exemplified by the advent of DOCSIS 3.0, serves to enable further disaggregation of the bundled model for video consumers.

As the operators' service delivery platform continues to change, however, so does the software applications platform. The availability of cable-delivered EBIF [Enhanced Binary Interchange Format], tru2way and unicast streaming applications such as NDVR [network-based digital video recording], will allow operators to more easily introduce new video services – subscription, transactional, and advertising supported. The debut of a new generation of "broadband-connected" TV sets will catalyze this as consumers will find it easier to "watch television" without subscribing to the traditional operators' services.

In the same way that the Internet is shifting toward a "cloud" model -- where computing resources are provided as a service, over the Internet -- so can multichannel video providers leverage their local, regional and national footprints to be the "TV cloud" for more and better content.

Specifically, this approach could deliver a stream of personalized content as a single, standard MPEG stream -- to any digital set-top box or web-connected CE device.

This paper provides an overview of current impediments to TV-based interactivity, and describes the high-level requirements and capabilities of the "cloud computing for the TV" architecture that could improve the outlook.

## THE CASE FOR THE CLOUD

The language of the computing "cloud" is typically associated with the Internet, even though the term itself pre-dates the Internet by at least two decades. Before it was the "cloud," computer scientists recognized the need for "grid computing," "distributed processing," and other variations on the theme of sharing a processing workload over clustered computers.

The benefits of cloud computing apply to interactive television, as well. The "TV cloud," for instance, could remove workflow gaps and bridge the application and media processing requirements between head-ends and set-tops. A cloud approach also would enable developers of programming and advanced advertising to use familiar, web-based development tools that are similar to or the same as those used to provide interactivity within a web site (e.g., DHTML, JavaScript, etc.).

As in all approaches, there are pros and cons to the notion of locating computing and media processing power in the operator network, versus investing in generations of more powerful Customer Premise Equipment (e.g., set-top boxes). Noting that each user is given an individual, personalized video stream can sum up the con of the network approach. Obviously, this uses the operator's network bandwidth to carry the unicast stream. However, the pros of this approach include:

- Faster software application lifecycle and lower total cost of ownership (development, integration, deployment, content management).
- The ability to deploy rich media services (i.e., complex video, graphics, and interactivity)to all CPE, eliminating "lowest common denominator set-top issues" and increasing scale for operators, programmers and advertisers.
- Consistent presentation of content and advertising opportunities (which minimizes unpredictable nature of the user experience on different CPE and optimizes the ability to accurately place advertising elements).

There are certainly capital investment attributes to the argument above. Generally speaking, investment in server-based computing and network bandwidth leads to less expensive CPE being able to provide an array of video services (i.e., simple MPEG decoders). That said, CPE with more capabilities can be equally supported. With respect to the investment in software technologies, the greatest expense often is in targeted development, integration and regression testing across dozens of different CPE platforms -- each with its own performance characteristics, graphics display capabilities, and consequent impact on the viewer experience. While tru2way, EBIF, and other standard client software platforms are certainly an improvement, they do not eliminate the cost of integration and regression testing.

As cable operators and multichannel video providers have deployed more and more sophisticated applications, the costs of client development have increased greatly – measured both in the time to market (24 to

36 months not being at all uncommon) and direct technology expense for complex labs and software expertise.

The server-centric approach allows an application to be developed, integrated, and regression tested on a single software platform. The time to realize a new or modified application can be measured in months (if not weeks). The server-centric approach takes advantage of powerful computing resources to enable sophisticated, media rich applications that can meet the operators' collective need for a rapid and repeatable way to "change the category," by creating must-have features that are unique to the video platform.

Given the growth in IP bandwidth – both the access and backbone networks – delivering "TV Quality IP Video" is quickly becoming a viable model. The server-centric approach is well suited to deliver video services over the IP network (somewhat obvious when one considers the growing number of such providers).

Questions for operators relate to development and deployment of new video services that maximize their investments in their robust, high QoS "last mile networks" for both traditional HFC and IP connected devices. What is the appropriate application software model for video applications? In short, via "the server cloud or the CPE [set-top] client?"

Addressing the industry's set-top fragmentation problem – meaning the hundreds of permutations of differing hardware and software combinations that exist amongst the 37+ million digital cable boxes currently installed – will go far in assuaging the "critical timing factor" necessary to quickly remove applications that aren't attracting consumers, and to quickly add those that are.

In addition, the total cost of ownership for applications lifecycle is a phenomenon well understood in the enterprise and personal computing domains, but only recently becoming an issue for multichannel video providers. In this sense, "total cost of ownership" includes not only initial applications development, but also customization, integration, bug fixing, and regression testing. That is, a "fat client solution," to the exclusion of all else, is an extraordinarily difficult and expensive business proposition.

One of the features of the Web is that it is essentially a "cloud computing" model in the "software as a service" sense. It allows the application lifecycle to be managed on a small and well-behaved domain of common devices – network servers and common client devices, i.e., PCs. The fact that PCs all have enormous computing capabilities means that media-rich applications depend on client-based media processing technologies for application execution (e.g., Flash).

Multichannel video operators can share the benefit of the Web approach and utilize network-based servers to support complicated, media-rich applications -- but they cannot depend on the client to have the necessary capabilities. Unlike the ubiquity of just one or two operating systems and common APIs, the multichannel video provider has dozens, if not hundreds, of client [set-top] platforms - all differing in the details of capabilities and performance.

Standardization of the client side will take many years. By analogy, it took more than a decade for Microsoft Windows® to become a de facto standard for desktop PCs. Application deployment to PCs is a demonstrably slow and expensive proposition as compared to the "web model" of software as a service. Web applications are iterated frequently and at low cost. Cable operators all have one common “standard” in their client devices, i.e., MPEG decoders. This, combined with server-based applications in the cloud will bring a more “web like” application life cycle model to the television viewing platform.

This Web model approach would combine the concept of “cloud computing” with some aspects of the traditional set-top approach. In essence, it binds together the best features of cable, cloud, and client.

## CURRENT IMPEDIMENTS TO TV-BASED INTERACTIVITY

The storied past of TV-based interactivity goes back as many as three decades, yet “interactive TV,” as a category, seems always to be on the horizon. The word “interactivity” itself is part of the problem: As soon as an “interactive application” gains consumer traction, it takes on a descriptor other than “interactive.” Consequently, the application exits the perceived realm of “interactivity,” becomes part of the “normal” viewing experience, and ceases to exist as an example of “interactive TV.” Examples include the electronic program guide (EPG) and video on demand (VOD). Both provide interactivity, yet neither is considered an “interactive application” as they have passed into the realm of the “normal” viewing experience.

The historical predominance of “destination-based” interactivity is also a factor. To access VOD, consumers were taught to “go to” a place to find and order titles. Likewise for the electronic program guide, which exists as a separate menu destination. A strong argument can be made for interactivity to occur as a natural part of the viewing experience that it enhances.

Immersive interactive video applications will bring the desired content “to the viewer,” not make the viewer search to find a “destination” in an unnatural way. This “surfacing the content to the viewer” (versus “destination-based interactivity”) can be found on many video streaming websites, e.g., YouTube, where the activity of viewing any given video stream is augmented by meta-data links to several other video assets (as well as non-video applets).

Transposing this experience to a full-screen video monitor, viewed at distance, and eliminating the computer mouse and keyboard bring the “VOD” and “interactive TV” models together as natural features of “watching television.”

## THE WORKFLOW ISSUE

Several other impediments to TV-based interactivity exist. One is an overall lack of an automated systems infrastructure to connect the “sales order process” with the “creative process” to the “content management and provisioning process” and finally to the “delivery process”. The overall word to describe this is “workflow.”

For the traditional multichannel video subscription business, this workflow is well established. In its simplest form, movies and TV shows are produced, licensed to an aggregator (e.g., NBCU), wholesaled to an operator (e.g., Comcast) for distribution, and then retailed to the consumer. The advertising and subscription models are well established for this process.

The important point is that there are automated systems (encoding, content protection, “billing systems,” trafficking systems, royalty payment and settlement) that support this model so these businesses can scale.

With respect to interactive applications, this “workflow” does not exist in any uniform, scalable way. The current ecosystem of extant and desired interactive video applications and services relies on a patchwork of business systems and creative tools, all of which are delivered to a heterogeneous population of operators with no “billable event tracking” except by sneaker-net and swivel chair operations. Without the “back-end” tied to the “front-end” via an automated workflow that generates invoices and tracks payments and respects copyrights, it will be very hard to build a scalable business around a popular interactive application.

A specific example of the workflow conundrum is the notion of the “bound” application, meaning an application that executes synchronously with the program or advertisement within which it runs.

EBIF, the Enhanced Binary Interchange Format, is the Cable Television Laboratories specification developed to establish bound applications over two-way video plant. EBIF’s strength is its overall reach – essentially the entire installed base of digital cable set-tops. However, EBIF defines only a portion of how to execute bound [program-synchronous] and unbound applications.

Specifically, EBIF defines only the delivery chain of the “trigger” or “widget” that enables a consumer to “click” from the remote control and to engage with the TV and the program or advertisement at hand. Such definition is critical and necessary, but for EBIF-based “bound” applications to become mainstream, a necessary scaffolding of workflow must emerge.

That workflow scaffolding includes the following: A known, easy and repeatable method for creating applications and applying any QA [quality assurance] mechanisms to ensure applications behave at their best; data collection, to fulfill the application’s intent, and to feed any primary or third-party billing mechanisms; and the links to those billing systems.

Consider an advanced advertising application that allows the viewer to click on a widget associated with an ad to receive more information on the product. From a workflow perspective, gaps emerge immediately:

1) *Creative*: What should the widget look like? Who builds the creative for the campaign – and to what template, using what authoring tool(s)?
2) *Application provisioning*: Operationally, the interactive application must be provisioned on to the network – its widget assets transferred for playout, its availability parameters fed into the traffic/billing system.
3) *Stewardship*: All ad campaigns follow general and specific rule sets – competing products may not be shown within the same ad pod; time parameters to protect children from inappropriate content, etc.
4) *Data Collection*: After playout, data associated with the spot needs a method to flow into the aggregation engines feeding national and local campaigns.
5) *Billing*: Any additional revenue associated with the interactive spot needs a feed into operator billing systems.
6) *Reporting and Settlement*: automated mechanisms must be available to

operators and advertising constituencies, etc. to create reports both for advertising effectiveness and contract fulfillment purposes.

While many efforts are underway, the authors do not know of any available solutions that will connect the traditional day-to-day business of advertising sales to the operators' broadcast and unicast streaming platforms. Individually and combined, workflow gaps prevent the business from scaling and the ability for multichannel video providers to build both local and national advertising revenues.

## THE CHALLENGE OF THE INSTALLED BASE

Digital cable set-tops, as a category, are approaching their 15th anniversary. Until fairly recently, they've existed as the "thin clients" that lag the Moore's Law trend of computing devices. Compared to PCs, digital set-tops have long been dismissed as not including enough processing power or memory to enable immersive, media rich applications. In short, what's thick today is thin tomorrow, and, for digital cable boxes -- compared to PCs -- it's always tomorrow.

The installed base of digital boxes presents a "lowest common denominator" problem for application development and software version control. Building applications only for high-end boxes reduces potential reach; building applications for all set-top variations reduces the application's attractiveness to the lowest common denominator of graphics chips, processing power, and memory.

Put another way, operating interactive applications solely upon the limited capabilities of the aggregate set-top base, and without the benefit of network server resources means the wealth of capabilities in the newest units is eclipsed by the care-and-feeding needs of the oldest units.

## THE CHALLENGE OF THE INSTALL-*ING* BASE

Equipment fragmentation problems are not contained to the set-tops of multichannel video providers. While the "TV Widgets" so prevalent at the 2009 Consumer Electronics Show achieved high marks for "cool factor," they will ultimately face similar challenges.

Assert: Consumers will tolerate poor quality of service when using an "interactive application" on the Internet versus on the TV. Consider: Millions of dollars have been spent trying to make channel changes take half a second less time, because consumers dislike having to wait two seconds instead of one and a half to change from one video stream to another.

Consider the developer, though, seeking to get a new "Yahoo Widget" into a Samsung television set. First, the widget gets submitted to Yahoo's TV widget working group for a testing cycle (with associated costs) that may be lengthy. After approval, the widget is presented to Samsung, likely requiring a convincing business model agreement to justify association with the Samsung brand. These steps are repeated for inclusion on other "widget-ready" devices, and likely would not be possible for an independent widget developer to perform.

Another popular client environment, Adobe Flash®, typically releases an entirely new software stack on an annual basis. Each release assumes a Moore's Law-like increase in available computational power and memory in the underlying platform –

yet that assumption likely will not fit the razor-thin margins and cost justifications required in the CE manufacturing environment.

Indeed, making a Yahoo Widget or Adobe Flash engine run exactly the same, on all consumer display devices, and so that the application author is not exposed to a maze of confusing, conflicting specifications is a very complex task. The result, as has been the case with set-top boxes, is the likely pruning of platform features to the lowest common denominator.

## REQUIREMENTS TO ENABLE SERVER-CENTRIC APPLICATIONS ON THE ACCESS NETWORK

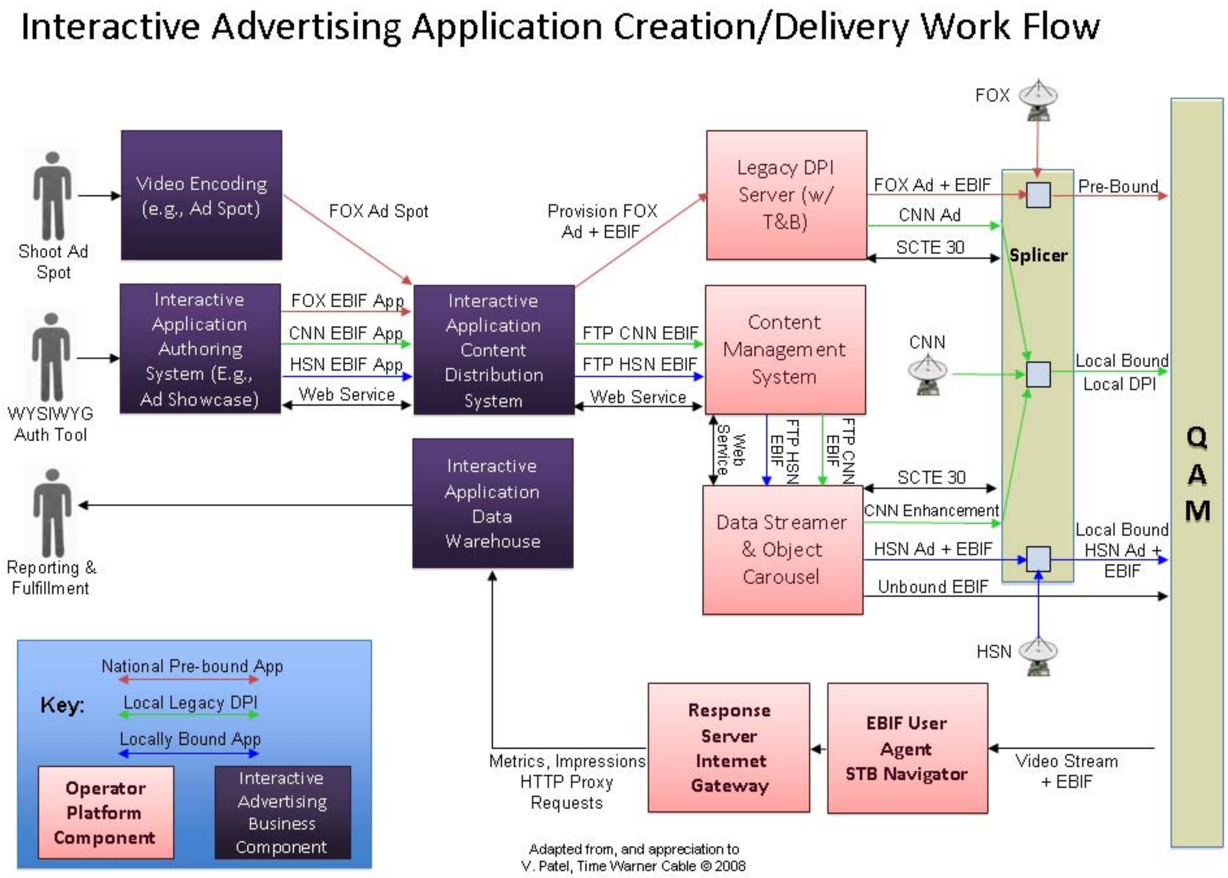
The “new new thing,” vis-à-vis using server-centric applications to deliver interactive video applications to thin client set-tops and/or CE client devices, is connecting the population of client devices to the server cloud with sufficient bandwidth. This seems obvious yet, in practice, it more than likely is not. Why: Multichannel video operators generally maintain copious amounts of bandwidth to connect their IP access networks to the Internet,  yet the traditional headend is not likely to have more than a data connection for File Transfer Protocol (FTP) support. This is a result of the fact that the subscription video content has traditionally been served by point-to-multipoint satellite distribution, or by low bandwidth links for IPG data or other non-video applications.

Requirements to enable server-centric applications, using content cached locally as well as pulling content from Internet sources, include:

Requirement 1: Provisioning of backbone network interconnections to 100 Mbps or greater between headend servers and relevant Content Distribution Network locations. This is obviously sensitive to the characteristics of individual applications. In effect, operators need to enable a terrestrial caching point at the edge of set-top network (as is done for IP services). The required bandwidth will be a calculated value derived from customary traffic analyses.

Requirement 2: The provisioning of adequate access network bandwidth to deliver unicast video streams to a meet a specified QoS. That is, the operator will estimate “peak simultaneous stream usage” and ensure the network Session Resource Management components are provisioned accordingly.

Requirement 3: The transcoding of content to formats supported by the client population. While there is a diversity of formats on the Internet (and a diversity of plugins to process them), the video streaming format that ALL CPE devices can process is MPEG. In addition, for cable operators, the ability to multiplex EBIF applications and/or deliver Java applets is required.




<u>Requirement 4:</u> Workflow support for integrating the application platform with "business systems" and "content management systems" back ends. This issue was outlined in the workflow discussion.

<u>Requirement 5:</u> The availability of common Web authoring. That is, tools that use DHTML as well as common scripting and video object toolsets. As applications are transcoded to the specific client capability (as above), the application developer does not need to be schooled in the various client capabilities and idiosyncrasies.

<u>Requirement 6:</u> The guarantee that applications will present the user experience per the designer's intent, consistently, on all client devices

## DOWNSIDES OF A "PURE CLOUD" TOPOLOGY:

To be fair, an entirely cloud-based approach to interactive TV isn't the right answer, either. Doing so wouldn't take advantage of client device capabilities, like local graphics blending, and a location execution environment for "overlay applications." An easy example of this is the ability to detect embedded data "triggers" on broadcast video streams (think "EBIF Trigger"). This allows a local application to "overlay" graphics on the video stream (and execute application logic) only for those viewers who choose to do so; it creates a locally interactive individual experience with a broadcast video service. There is no need for such a mechanism with unicast video streams as the viewer is already engaged in an individual, one-to-one, experience. As the bandwidth intensive video is shared (broadcast), this approach can be more efficient than unicast streams.

Likewise, certain applications are almost entirely client-based. Chat and social networking applications are good examples. They typically require a user to log on and the network to maintain knowledge of their location, a.k.a., "presence-based" applications. These applications are typically comprised of low-bandwidth text and graphics and are not associated, or bound, to a unicast video stream. Simple text and graphics can usually be processed by client devices using graphics overlay and are not directly associated to a unicast video stream (think "chat room associated to a live sporting event").

The ideal environment for immersive, video-linked interactivity, which infuses Web-like characteristics into TV shows and ads, is a combination of cloud computing and a traditional client-server applications architecture approach. Network servers have the processing power and resources to create media-rich, video-intensive applications. Client-side application execution can provide associated overlay applications, as well as detect embedded video triggers that can then be locally processed (on the client) to further enhance the viewer's experience by offering hyperlinks to related video or other applications.

An example of this could be a TV commercial with embedded EBIF triggers that enable telescoping into an ad microsite. The user reacts to the trigger, initiating a local application overlay. The overlay application displays several options, including the ability to see special how-to videos. The user selects a video of interest and the local EBIF application signals a network server, which initiates a unicast video stream in concert with the local application. The user is now seamlessly viewing a linked video. This video stream is inherently personal (as it is unicast) and may contain further embedded triggers. As the server can process user requests for complex media types and encode them to client compatible media formats in real time, the application may bring all the depth of a typical multimedia web site application to the viewer.

Meanwhile, a scaled version of the broadcast stream can be included in every scene, enabling the viewer to continue to view the original programming channel. Given appropriate "TV production values," the net effect on the viewer is no obvious "interactive" or "VOD" application, simply viewing a video with all the richness

Example of EBIF Triggers and Telescoping to an Ad Microsite

1. TV viewer is watching Trading Spaces, a linear TV program about home remodeling on The Learning Channel, when a two minute ad break begins


2. In the last spot, a Home Depot commercial airs. An EBIF on-screen prompt then appears giving the viewer a chance to see do-it-yourself videos


3. After pressing "OK", the viewer telescopes to a Home Depot-branded microsite featuring do-it-yourself videos, product demos and other features. Meanwhile, a scaled version of the linear broadcast plays in the bottom right corner


4. The viewer could watch do-it-yourself videos while learning more relevant products such about Black & Decker tools and Benjamin Moore paints


5. Home Depot can even choose to provide dedicated multi-layered showcases for each featured brand, effectively creating "microsites within a microsite"

6. At any point, the viewer could exit the Home Depot microsite and return to the full resolution linear broadcast

of the Web and the immediacy of the classic television presentation experience.

## CONCLUSION

The optimal solution to provide the best viewing experience that combines the media richness of the modern Web with the quality of service and ease of use of traditional television uses a software applications platform that maximizes both server-centric computing and media processing power with client overlay and local application logic. The Web model, if used as a guide for modernizing and realizing the promise of the cable broadband television platform, would argue to use authoring tools and publishing workflow for applications targeted at both mainstream video and advertising applications (the latter, taking advantage of the oft cited "targeting" capabilities inherent in the Web model and when using unicast video applications).

The judicious use of server-centric software applications will ameliorate (but not completely eliminate) the problem of lowest common denominator set-top fragmentation. This is a serious issue when considering "scale." The server and client hybrid – one that maximizes the inherent strengths of a high-bandwidth, real-time, two-way connection between the headend and home, and that takes advantage of simultaneous MPEG and IP transport paths -- is the optimal approach to maximize the return on investment in software delivered multimedia applications.