OpenCable™ Guidelines

Enhanced TV Operational Guidelines

OC-GL-ETV-OG-V02-091223

RELEASED

Notice

Document Status Sheet

Document Control Number:	OC-GL-ETV-OG-V02-091223			
Document Title:	Enhanced TV Operational Guidelines			
Revision History:	V01 - released 7/14/06			
	V02 - released 12/23/09			
Date:	December 23, 2009			
Status:	~~Work in Progress~~	~~Draft~~	Released	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	Public

Trademarks

Contents

List of Figures

List of Tables

This page left blank intentionally.

1 SCOPE

1.1 Introduction and Overview

The cable industry has defined a platform that provides a nationwide footprint for enhanced television. The platform definition includes specifications for application signaling and synchronization, application formats, and User Interface guidelines. This document describes an end-to-end architecture and a set of operational guidelines that programmers and MSOs may follow in order to deploy enhanced services.

The OCAP platform also provides a means for deploying applications that are delivered and executed in conjunction with television programming. The ETV platform provides similar functionality for limited capability devices that do not support OCAP. ETV applications can be supported on OCAP systems, but express a small subset of the features and functionalities of OCAP applications.

1.2 Purpose of document

The purpose of this document is to describe an end-to-end architecture and a set of operational guidelines that programmers and MSOs may follow in order to deploy enhanced services. These guidelines provide a framework in which the programming community and MSOs can cooperate to launch an enhancement on a nationwide basis.

Much of the value of enhancements to programmers is the ability to reach a massive, nationwide audience. This entails that enhancements be supported by multiple MSOs in a manner that does not require tailoring the formatting, packaging, or delivery of enhancements for specific networks. These guidelines provide an interface between the programming community and the set of national cable networks.

1.3 Requirements

This is a guidelines document. It is for informative purposes only, and does not place normative requirements on products. There are no plans to develop product conformance tests based on information in this document.

2 REFERENCES

2.1 Informative References

The following table enumerates the informative references used by this document.

[ETV-BIF]	OC-SP-ETV-BIF1.0-I05-091125, OpenCable Enhanced Television (ETV) Binary Interchange Format 1.0, November 25, 2009, Cable Television Laboratories, Inc.
[ETV-AM]	OC-SP-ETV-AM1.0-I05-091125, OpenCable Enhanced Television (ETV) Application Messaging Protocol 1.0, November 25, 2009, Cable Television Laboratories, Inc.
[ADI 1.1]	MD-SP-ADI1.1-I04-060505, CableLabs Asset Distribution Interface Specification Version 1.1, May 5, 2006, Cable Television Laboratories, Inc.
[HTTP1.1]	RFC 2616, Hypertext Transfer Protocol – HTTP/1.1
[SCTE 30]	ANSI/SCTE 30 2009, Digital Program Insertion Splicing API
[SCTE 35]	ANSI/SCTE 35 2007, Digital Program Insertion Cueing Message for Cable
[SCTE 67]	ANSI/SCTE 67 2006, Digital Program Insertion Cueing Message for Cable – Interpretation for SCTE 35

2.2 Reference Acquisition

CableLabs Specifications:

- Cable Television Laboratories, Inc., 858 Coal Creek Cr, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax 303-661-9199; Internet: http://www.cablelabs.com

RFCs:

- Internet Engineering Task Force (IETF) Secretariat, 46000 Center Oak Plaza, Sterling, VA 20166,
  Phone +1-571-434-3500, Fax +1-571-434-3535, http://www.ietf.org

SCTE Specifications:

- SCTE Headquarters, 140 Philips Rd., Exton, Pa 19341-1318;
  Phone +1-610-363-6888; Fax 610-363-5898; Internet: http://www.scte.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Return Channel	Cable Out-Of-Band (OOB) or DOCSIS Set-top Gateway (DSG) signaling path that transports messages from a receiver to a headend

4 ABBREVIATIONS AND ACRONYMS

DOCSIS®	Data-Over-Cable Service Interface Specifications
EBIF	Enhanced Television Binary Interchange Format; see [ETV-BIF]
EISS	ETV Integrated Signaling Stream
FOD	Free on demand
Legacy OOB	Legacy Out-of-Band
RC	Return Channel
RTF	Required Test Facility
STB	Set-top Box

5 CABLE NETWORK DISTRIBUTION

This section describes the transmission pipeline for delivering enhancements and their associated resources, and bi-directional communications between server and client modules.

5.1 Downstream

5.1.1 Television Distribution Organizations

The distribution of television signals has evolved in the United States into the creation of four types of broadcast organizations:

- National Networks
- Cable Networks
- Affiliate Stations
- Local Cable Stations

ETV distribution is discussed in detail for each of these entities, both as originators of ETV and has parts in the distribution chain.

The Broadcast Television system is a symbiotic relationship between the Nation Networks and their local Affiliate Stations. A National Network (ABC, CBS, FOX, NBC, WB, UPN) is an organization whose purpose is to commission the creation of television programs, sell national advertising, and distribute programming that is aired at specific times of the day on local Affiliate Stations around the country. The pooling of resources at the Network level reduces the cost of show production per Affiliate and allows for significantly larger show budgets for Network shows. The Affiliate Stations broadcast an RF signal that can be received by local residents, cable operators, satellite operators, and IPTV operators. Affiliate stations produce local programming that is aired in-between Network programming and are allowed to insert local commercials into special slots during commercial breaks of the Network programming. Most National Networks uplink their "network feed" to the local affiliates out of New York via Satellites. The local Affiliates receive the feed and either broadcast the feed live or tape it for playback later in the day. Affiliates in the Eastern and Central time broadcast live off of the ET/CT Network feed. Pacific Time zone Affiliates broadcast live off of the PT Network feed. Those Affiliates located in Mountain Time zone usually either tape the Eastern feed and then replay it at the appropriate time or broadcast the Pacific feed live. Some networks allow their MT Affiliates to insert an additional minute of commercials anywhere in the broadcast in order to compensate them for the cost of taping the ET/CT feed.

Cable Networks are national broadcast organizations that originally distributed their content through cable systems (though many now also distribute via Satellite, IPTV, and wireless). Most cable networks have two feeds for the country: East and West. The East feed covers the Eastern and Central time zones and the West feed covers the Mountain and Pacific time zones. Most Cable Networks distribute their signal via a Satellite to the MSOs though some may have fiber connections to nearby MSOs. The MSOs are allowed to insert "local cable commercials" at specially designated times during commercial breaks of the Cable Network broadcast.

Local Cable Stations are organizations who broadcast exclusively to a subset of the nation, usually only to MSOs in the local area. Their connection to the local MSOs is usually over fiber. MSO "local cable commercials" may or may not be inserted depending on the station ownership and content.

Support for ETV by cable programmers, affiliates and MSOs is fairly straightforward, given their investment to support ETV as their own service. These scenarios are described below. Support by national broadcasters presents challenges, as presented immediately below.

5.2 Broadcast Network Sourcing

This section describes in detail NTSC and ATSC signaling path options to support ETV.

5.2.1 Television Distribution Standards

In the United States, broadcasters currently are distributing video content in two formats: NTSC and ATSC. NTSC (National Television System Committee) or analog is the evolving standard that television has been utilizing since 1941. NTSC was originally a black and white format and electronic color was added in 1954. ATSC is the new digital standard for the transmission of HD television content and is based on MPEG-2. The broadcast paths for each of these standards are significantly different. An overview of these standards is in the Matrix below.

Table 5–1 - Matrix of ETV Distribution Scenarios

	National Network	Affiliate Station	Cable Network	Local Cable Station
NTSC (Analog)	Analog or Digital via Satellite	Analog via RF or Fiber	Analog or Digital via Satellite or Fiber	Analog or Digital via Fiber
ATSC (HD)	Digital via Satellite	Digital via RF or Fiber	Digital via Satellite or Fiber	Digital via Fiber

Reviewing the NTSC line in this Matrix, it becomes clear that NTSC can go through a number of Analog to Digital conversions before the signal arrives at the cable MSO. On the other hand, ATSC is Digital for all situations but may be converted back and forth from MPEG-2 to Baseband Digital Audio Video.

5.2.2 Broadcast Network basic signaling paths

The following diagrams illustrate signaling paths for NTSC and ATSC signals, without support for ETV. These diagrams provide a basic blueprint from which options to support ETV are described.

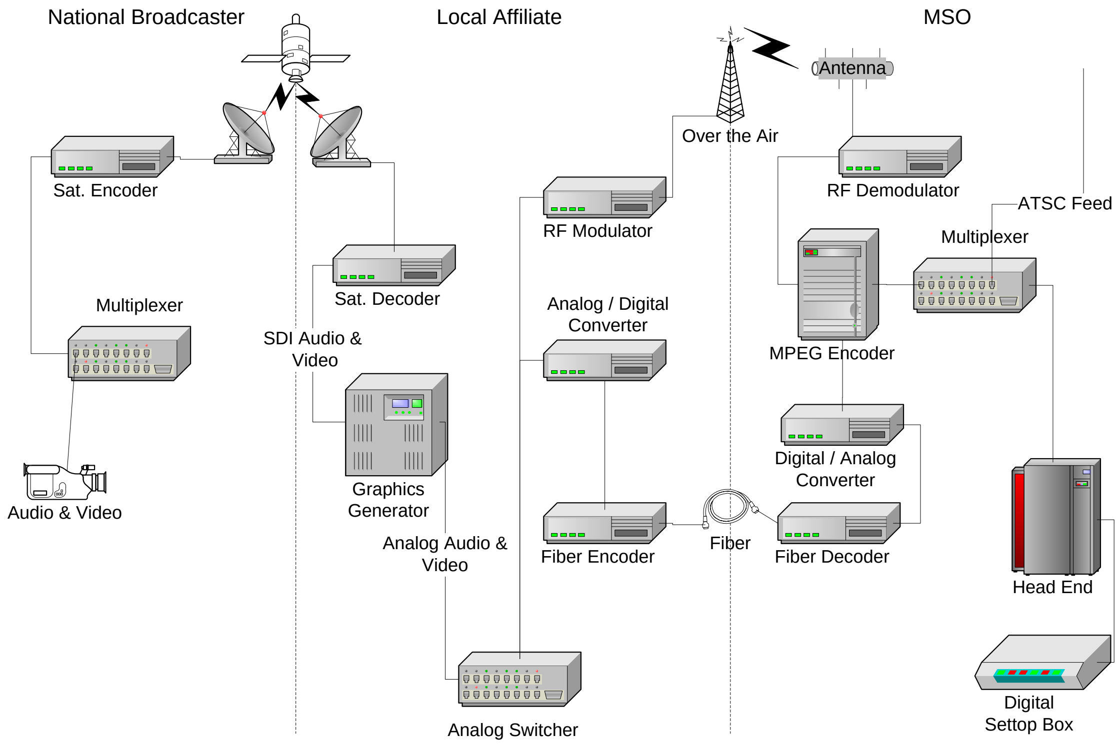

Figure 5–1 - Broadcaster Signaling Path – NTSC

Many MSOs receive their NTSC feed over-the-air via analog RF transmission. Since only triggers in the VBI would survive through such a transmission, delivering complete ETV through this path is impractical.

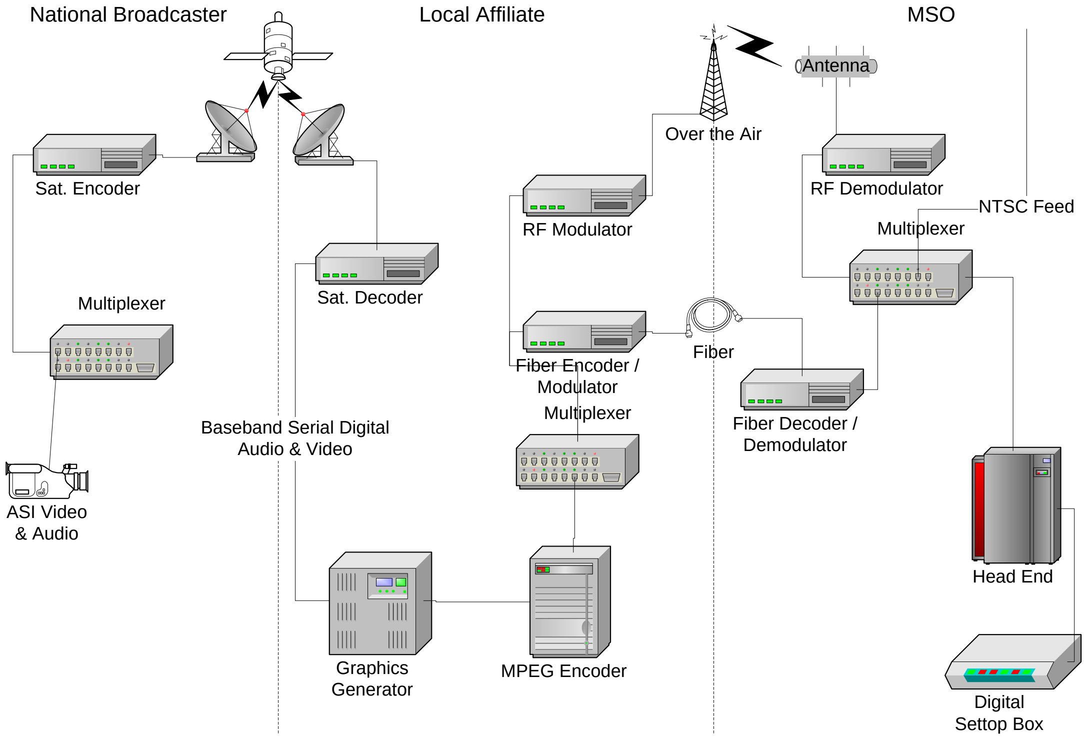

Figure 5–2 - Broadcaster Signaling Path - ATSC

The ATSC ETV signaling path is all digital but it is not MPEG-2 all the way through. FOX and the FOX Affiliate Stations are the only Broadcast Network that uses MPEG-2 splicers at the affiliate stations to insert local graphics. As a result, FOX is the only national network positioned to distribute ETV nationally without additional support. The rest of the Network Affiliates convert MPEG-2 to baseband serial digital audio and video for graphics addition and then re-encode the baseband serial digital audio and video into MPEG-2. Everyone but FOX is going to have to carry ETV through some extra transport mechanism.

5.2.3 Proposed Broadcast Network Solutions

Several approaches have been identified to deliver ETV applications associated with Broadcast programming:

- Affiliate pass-through
- National ETV feed
- Internet distribution to MSOs

Each method is described below, followed by an analysis of the pros and cons of each.

5.2.3.1 Affiliate pass-through

The following diagram illustrates an affiliate pass-through strategy that might be used to distribute ETV. In this scenario, the affiliate station is upgraded to include equipment that strips ETV data out of an ATSC program before its converted to baseband for graphics composition, then adds the ETV data back in after the program is converted back to MPEG. This method uses the ATSC feed (which is all digital) to transmit ETV to the MSO, who then cross connects these data PID(s) to the NTSC feeds.

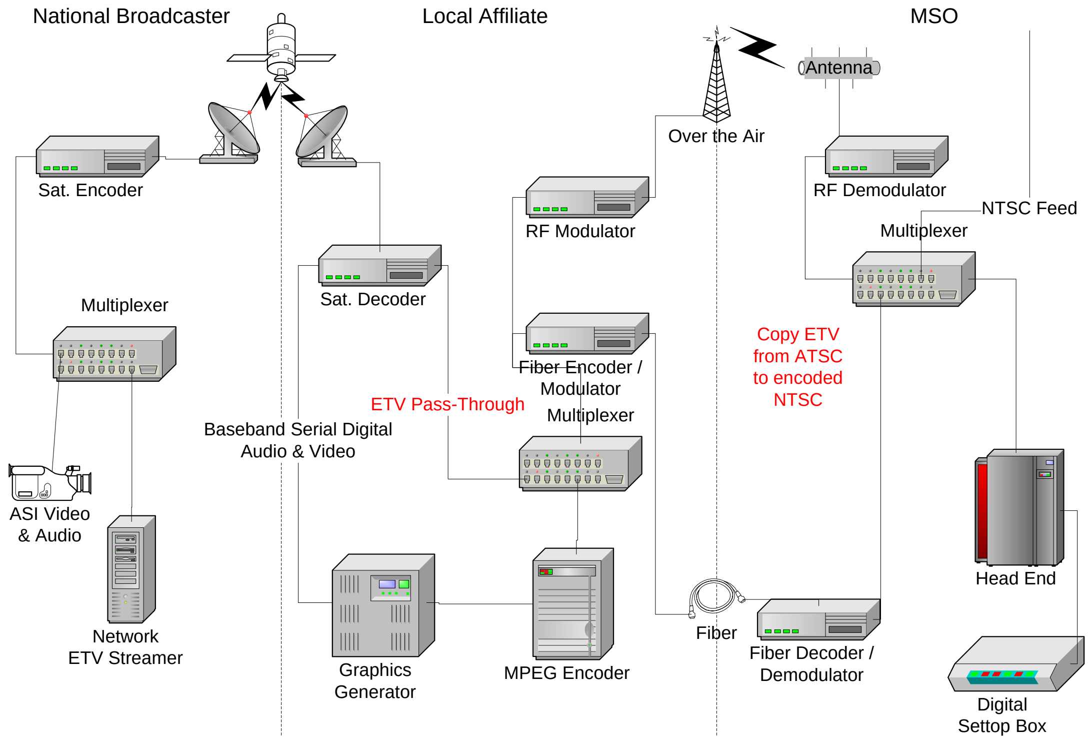

Figure 5–3 - Affiliate Pass-Through - ATSC

If ETV is to be delivered through the affiliates, the affiliate stations will have to agree to broadcast an estimated additional 1.5 Mbps ETV data in their RF feed. For ABC and FOX, this is less of an issue since both networks are using 720p which can be safely compressed to 16 Mbps of the 19.4 Mbps of bandwidth available for transmission. CBS and NBC affiliates are utilizing 1080i which require significantly more bandwidth (approx. 18 Mbps) so it is unclear where they will get the extra bandwidth to broadcast ETV over the air.

Requiring all of the Broadcast Affiliates to implement the EBIF Pass-Thru is not practical and would require a significant investment by each Affiliate station. Where NBC or CBS would get the additional 1.5 Mbps in their ATSC feed is also an issue. At the MSO, every MSO would have to invest in hardware to strip out the EBIF data from the ATSC feed and reinsert this data into the NTSC feed.

A variation on this theme is that the affiliates change their ATSC processing path to composite graphics into MPEG content.

5.2.3.2 National ETV feed

To avoid the burden of upgrading affiliates, a nationally broadcast ETV feed that is received directly by cable offers an alternative. Here the broadcast networks create a separate satellite feed per time-zone for ETV. MSOs would mix at the headends the appropriate ETV satellite feed with the local Affiliate A/V signal. The following diagram illustrates this method.

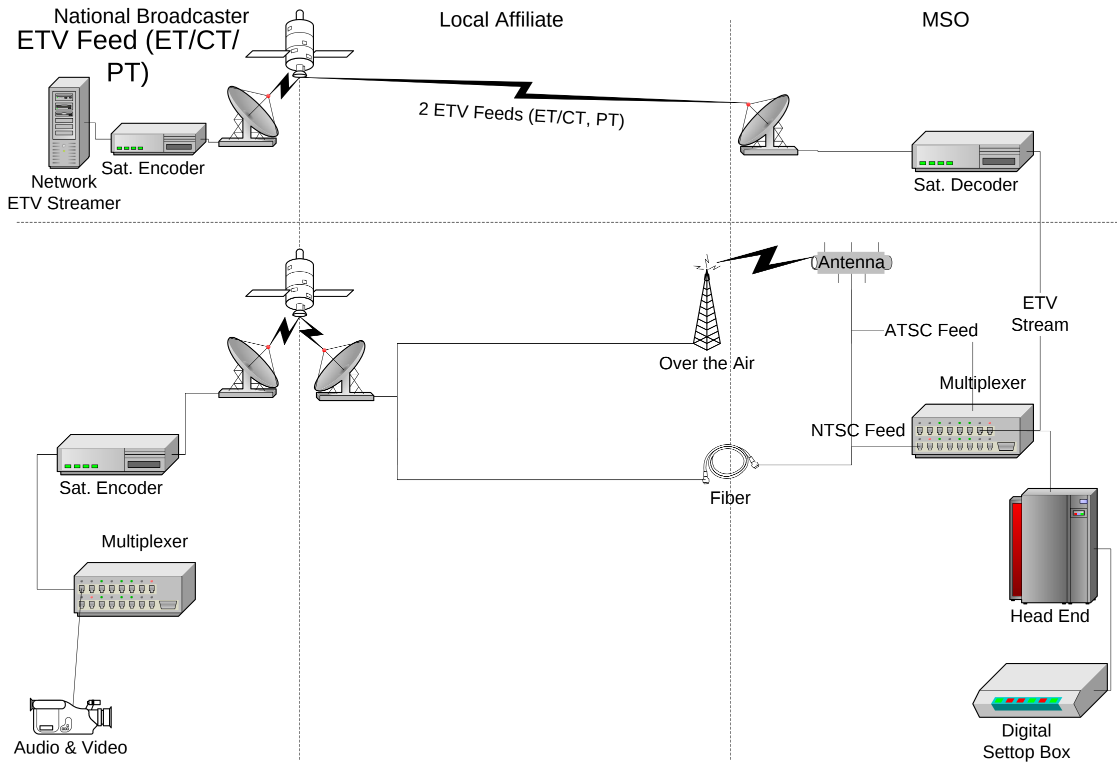

Figure 5–4 - National ETV feed

This solution presents a set of challenges. Mountain time-zone affiliates generally tape delay programming, which introduces synchronization issues, and a means by which affiliates can influence the feed during disruptions to normal broadcasting is needed. There is also a scenario where the affiliate adds ETV to the national feed. These scenarios are described below.

5.2.3.3 Internet distribution

This method proposes adding an ETV 'streamer' at each MSO headend, which can receive ETV content from broadcasters via the Internet.

Here MSOs install local ETV streamers at headends that have ETV loaded on them, via pull or push. Use Internet connectivity to the local streamers to transmit triggers.

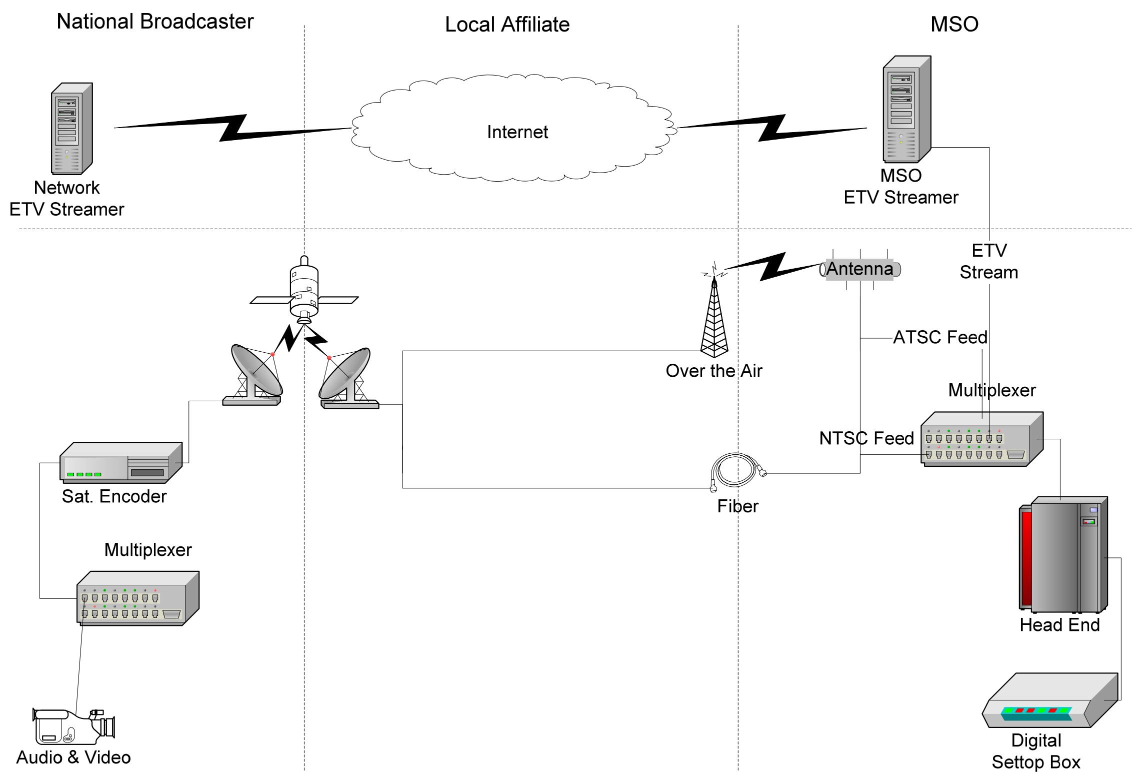

Figure 5–5 - Internet Distribution of ETV

Installing MSO local streamers at every headend is costly and creates legal issues over service delivery between the Broadcasters and the MSOs.

Synchronization between enhancements and associated programming might be achieved through the use of scheduling data. Schedules could be propagated to MSO local streamers along with enhancements.

5.2.4 Analysis of Proposed Broadcast Network Solutions

Each approach presents challenges. Since each requires some investment in equipment, mixing the solutions across the national market is not efficient. One solution is expected to emerge as the sole solution.

Table 5–2 - Comparison of Proposed Broadcast Solutions

Solution	Pros	Cons
Affiliate pass-through	• no impact on broadcaster • maintains synchronization between ATSC and NTSC	• Requires investment by affiliates • Requires investment by MSO to copy ETV from ATSC to NTSC • Assuming <%100 adoption by affiliates, erodes nationwide availability • Only applicable to programs that are broadcast in ATSC and NTSC (~%30)
National feed	• no impact on affiliates (except MTN)	• requires investment by broadcaster in satellite resources • Lots of headaches to support MTN • Requires an affiliate over-ride mechanism

Solution	Pros	Cons
Internet	• No impact on affiliates • Inexpensive hardware	• Difficult to scale/maintain • Requires investment by MSO in ETV streamers • Synchronization is difficult

Assuming that a "Mountain Minute" time could be arranged, delivering ETV in separate Satellite feed to the MSOs seems like the optimal solution. The MSOs would use their existing Mux to combine the A/V feed from the Affiliates and the ETV feed from the Network. The ETV feed could be transmitted via the same satellite system as existing Cable channels use for distribution, thereby requiring minimal equipment changes at the headend. From the Affiliates' perspective, the "Mountain Minute" time would have to be consistent across all Affiliates in MT, but no additional hardware would have to be installed and no additional bandwidth would have to be allocated. At the Networks, ETV streamers would be installed and at least three additional satellite uplinks would have to be installed (one per time-zone). Timing between the ETV feed and the Network A/V feed could be tightly controlled and average delay times to the MSOs could be added to the streams to improve synchronization.

5.3 Local Affiliate Sourcing

Local affiliates adding ETV to their local programming; such as local news, local sports and local weather, may either inject ETV content into their local content or pass the ETV content to MSOs to inject it. Both scenarios require that the affiliate has equipment to act as an ETV streamer. These scenarios are described below.

If the national broadcast ETV feed model described in Section 5.2.3.2 is adopted, MSO equipment would have to have the capability of switching from the feed to the locally-sourced content when need be.

5.3.1 Affiliate injects ETV into programming

In this scenario, an affiliate injects ETV into their digital programming. In cases where programming is dual carried in NTSC and ATSC, MSOs will need to support the 'patch' solution described in Section 5.2.3.1, where the MSO copies ETV from an ATSC feed and pastes it into an associated NTSC feed when the NTSC program is digitized.

5.3.2 Affiliate sends ETV to MSO for injection

This scenario is a variation on the Internet distribution described in Section 5.2.3.3, where an affiliate's ETV streamer sends ETV data to an MSOs ETV streamer, which injects the ETV content into programs. Coordination and synchronization issues would have to be addressed by both parties.

5.4 Cable Channel Sourcing

Cable Channels have a significantly easier time delivering EBIF data enhancements because all of their content is in digital formats. All that is required is that the Cable channel Multiplex into their existing MPEG-2 signal the ETV content. Analog cable stations would have to upgrade to MPEG-2 distribution in order to support ETV enhancements.

5.5 MSO Sourcing

Local MSOs may have interactive programming that they want their viewers to access. Such programming may include local interactive advertising, community events and even local t-commerce. The signaling required for this scenario is quite simple as all assets, applications and content is available through the local operator, and simply injected into programming by ETV streamer equipment.

5.6 Application Management

This section describes the transmission pipeline for enhanced content and identifies control points MSOs may use to configure, monitor, and manage application delivery.

MPEG-2 Transport Streams (TS) may contain Data streams that carry application data. In the case of ETV, there is one Data stream allocated to EISS application signaling, and one or more Data streams allocated to EBIF application data. ETV defined Registration Descriptors within the PMT indicate the PID location of ETV Data streams.

Some cable set-tops, QAMs, and other headend systems only process a PMT when a broadcast service is acquired after a channel change. Because updates to a PMT will not be detected while tuned to a broadcast service, the presence of an application where there had previously not been one (and vice versa), will not be discovered. Any service that may include an application must therefore always include PMT descriptors indicating the presence of application Data streams. When no application is present in the service, the application Data streams indicated by the PMT descriptors will not be present in the program. When no application is present, the PIDs indicated by the ETV PMT descriptors may be called ‘placeholder PIDs’.

Placeholder PIDs may technically be considered illegal in some specifications (DVB?, SCTE 42?), and might cause warnings by some TS analyzers; however, they are already used in some environments and there are no known general problems with their use. SCTE 35 has an extremely low and bursty data rate, so it acts like a placeholder PID and is in widespread use today.

ETV User Agents (UA) are expected to be able to handle placeholder PIDs. If a PMT indicates Data streams, the absence of those streams is simply expected to result in a no-op by the UA.

Local ad insertion by service providers may splice application components s into a broadcast service in addition to video and audio components. The service feeding this splicing process must include PMT registration descriptors indicating the location of application streams. The PIDs with registration descriptors may be part of the source feed for the broadcast service, be placeholder PIDs created by an upstream device or be placeholder PIDs created within a splicer.

An application may be present in content even when there is no agreement authorizing presentation of the application with a given service provider. For example, a national broadcast feed containing an application may be distributed to a number of service providers, only some of which may choose or be obligated to present the application. Service providers must be able to selectively enable and disable propagation of applications in a content source within their distribution systems. One mechanism to disable application presentation is to remove the Data streams associated with an application.

Note: This section does not address application management by a User Agent. This topic will be covered elsewhere.

5.6.1 PID Management

Service Providers must be able to perform a number of operations on content streams in order to properly deliver interactive services. The essential operations that a service provider may perform are:

- Add ‘placeholder PIDs’ by adding PMT registration descriptors for ETV applications.
- Pass-through application Data streams and associated PMT registration descriptors.
- Remove application data from application Data streams.

5.6.1.1 On-Boarding

Before a service can support ETV, the service itself must be configured and the video distribution chain must be configured.

On-boarding of a given service addresses static configuration of placeholder PIDs or ETV data pass through.

5.6.1.1.1 Service Configuration

A service that contain ETV, inserted at any point in the distribution chain, must be configured to always indicate the appropriate Data stream in the PMT. This configuration is ideally performed by the content originator, but may be performed by a content aggregator or a service provider. Configuration means adding ETV descriptors to the PMT, one for EISS application signaling, and one or more for EBIF resource streams. All application components must be configured with the PID values indicated in the PMT. Note that the registration descriptors defined by different versions of the ETV platform, e.g. I03, I04 of [ETV-BIF], are slightly different. These differences may be important to some splicer implementations, where the version of the EISS and EBIF contents within a Data stream must match the version of the registration descriptors.

Because ETV services may contain more than one application resource stream, a service must be configured to indicate the maximum number of Data streams in the PMT.

5.6.1.1.2 Plant Configuration

Every piece of equipment within the video distribution chain must be configured to pass through the ETV service. This means leaving the ETV descriptors in the PMT, and passing through ETV Data stream when present. Leaving the PMT untouched is assumed behavior for all equipment. Enabling pass through of Data stream may require static configuration of certain equipment. In practice, each head-end must be configured to pass through the ETV Data. Proprietary tools make this job simple to perform, but it is a manual step that must be performed throughout a service provider's entire ETV enable footprint, for each enhanced service.

5.6.1.1.3 Change Control

If there is a change in the service (ex. expand the number of ETV Profiles, and hence Data streams, delivered from a content provider), the service and plan configuration steps described above will need to be repeated in part or in their entirety.

5.6.1.2 Stewardship

The general framework for application management assumes a 'stewardship' function implemented by a service provider. The stewardship function executes logic that performs application management. It has knowledge of Campaign Information Package (CIP) instances that are associated with authorized applications, and interfaces with distribution equipment. The stewardship function is a logical entity that may be implemented in different ways by service providers; the term does not refer to a set of equipment or specific workflows.

Distribution equipment is assumed to support proprietary interfaces that allow a stewardship function to manipulate PMT descriptors and streams within a program. Groomers and other equipment can generally filter streams by stripping a stream while leaving the PMT alone. In one approach, an MSO software process, driven by the stewardship function, could query a groomer or another piece of equipment, to discover the PMT of a given service. If the service contains an unauthorized application, the MSO software, using the PID values in the PMT, can direct the equipment to 'drop' the offending streams.

The stewardship function, driven by the CIP, has knowledge of certain attributes of authorized applications, such as AppID, PEID/EPSID, and service ID and flight window. Unless and until equipment is upgraded to understand application level types, like AppID and PEID/EPSID, the stream filtering described above must be implemented based on service ID and window. Window is based on wall clock, and therefore will not be frame accurate, and may become less accurate during overrun scenarios, such as when a ball game goes into overtime. More grave, service IDs are not universally defined across service provider systems. Until a single service ID scheme is adopted, there is no generally applicable way to filter applications based on CIP. ***Note***: This section does not address other topics related to stream management, such as bandwidth allocation and format conversions. These topics will be covered elsewhere.

5.6.2 Bandwidth Management

EBIF signaling and resources are Constant Bit Rate (CBR) data streams carried within an enhanced program. This represents overhead that must be accommodated in already fully-utilized video distribution systems.

5.6.2.1 Distribution Scenarios

There are three distribution scenarios that must be considered: Switched Digital Video (SDV), traditional video encoders, and statistical multiplexers (StatMUX).

5.6.2.1.1 SDV

In an SDV environment, a fixed bandwidth budget is allocated for each service. Data within a service must be allocated out of the overall budget for that service; in other words, the service must be encoded such that the video, audio, and data components do not exceed a well-known value. This is a nicely scalable solution, as each service regulates its bandwidth. A service that includes an enhancement trades off video bandwidth and those that don't enjoy the full bandwidth for video. Current SDV solutions employ Constant Bit Rate (CBR) encoding of video. They ensure that a given service does not exceed its bandwidth budget, re-encoding to the allocated bit rate, or rate-clamping the service on a StatMUX can be used. It is not anticipated that a service will exceed its bandwidth allocation.

5.6.2.1.2 Traditional encoders

In this scenario, a group of programs are multiplexed together into a single, fixed rate multi-program output stream by an encoder. Current encoders allocate separate fixed sized 'pools' for video and data. In essence, this requires that a certain data bandwidth be reserved for the entire multiplex. The collection of video streams is compressed to a certain level regardless of whether data is present or not. Encoders must be pre-configured, setting the sizes of the video and data 'pools' and the output bandwidth.

Newer encoders and transcoders take in compressed streams with data as input and can manage the video and data as a single pool similar to a stat-mux.

5.6.2.1.3 StatMUX

StatMuxes dynamically accommodate data overhead by trans-rating the set of video services within a multiplex, in effect spreading the load over the multiplex. A large amount of data overhead within a multiplex, either through a large percentage of enhanced services, or enhancements that contain large amounts of data, will adversely affect video quality across the multiplex.

5.6.2.2 System expectations

In general, initial nationally distributed applications, for instance those from Canoe Ventures, are expected to be encoded to not exceed 100 kbps, regardless of the total size of all application resources. A typical application might be 100 KB in total size, resulting in a maximum loading time of 8 seconds if encoded at 100 kbps. Larger applications will take longer to load. Eight seconds may be an intolerable time to load an application, in which case the total application size must be made smaller. In practice, simple but useful applications are often 20 – 25 KB in size. MSOs need to have foreknowledge of all enhanced services and the maximum data rate for each service in order to configure their systems accordingly.

5.6.3 Future Topics

5.6.3.1 Full and Advanced Profile applications

ETV defines Baseline, Full and Advanced profiles in order for applications to take advantage of capabilities on a wide range of device. For instance, low-res images might be transported for use on Baseline devices, and higher res images used for devices that support Full or Advanced profiles. Taking advantages of Full and Advanced profiles can greatly increase the size of an application. Extending the example, bitmaps are often the largest files with a set of application resources, so support for multiple profiles greatly increases the size of bitmaps. Given that load times often cannot be excessive, either use of Full and Advanced profile applications, and associated high-res bitmaps will be discouraged, or exceptions will have to be made to the 100 kbps convention.

5.6.3.2 MPEG-4

Given the greater video compression offered by MPEG-4 encoding, a fixed rate application data stream consumes a much greater percentage of the overall service bandwidth. Implications of this have not been explored.

5.6.3.3 Variable Bit Rate (VBR) Applications

If applications can be encoded in VBR format, it provides greater flexibility in the video distribution system, and may significantly increase the total number and bit rate allocations for applications.

6 RETURN CHANNEL

This section describes the enhancement interaction between client and server modules. The term **Return Channel** is abbreviated to **RC** from this point onwards.

6.1 Data Formats

In general a set of servers will be dedicated to support enhancements and will contain the server components that respond to requests from the enhancements. Return channel interactivity from client components will target these servers for both additional data feeds and application-specific transactional processing.

Various server roles are described below. However, one is ETV-specific and will be described immediately. This is the **EBIF RC Server**.

This server has the role of translating the response from an application server (that may be HTTP or HTTPS or other protocol, but not EBIF binary format) into EBIF binary format as expected by the enhancement on the set-top. The use of a dedicated server is optional and the application server handling the request can perform this conversion itself.

As described in [ETV-BIF], the request from the enhancements will be HTTP or HTTPS v1.1.

The requests will contain the form values collected by the enhancements as POST name-value pairs.

Reponses to the request as generated by the application servers will have to be EBIF binary format.

Further information can be viewed in the following sections of the EBIF specification:

Section 6.4.3.4.4	Form Submission Processing
Section 6.4.3.4.4.1	Form Response Entity Processing
Section 11.10	Form Items

Let us now consider other server roles.

6.2 Server Roles

Other servers in the proposed architecture are as follows:

- Content Servers—servers containing information and content in support of various enhancements.
- Transactional servers—servers that handle specific requests from potentially many applications
- Application servers—same as Content or Transactional servers above
- Authentication servers—servers to provide authentication of requests from set-tops, assumed to be handled by the headend
- Proxy servers—servers translating vendor-specific addressing schemes to IP address schemes, assumed to be handled by the headend. These should support direct socket connections to large broadcasters via RTSP or similar protocol.
- Database servers—servers for persistent data storage often supporting the servers above, can be seen as equivalent to the Content and Transactional servers above

- Operations and monitoring servers—servers for monitoring and management of enhancements, not described in this section of the document.

Validation of the addresses targeted by the enhancement on the set-top will not be carried out by the enhancement itself but will be performed by the headend network servers such as the network controller and proxy server.

6.3 HFC Network Details

This section describes several aspects of the connection between a set-top and the cable headend equipment.

6.3.1 Viewer Identification

Viewer identification is not described in this document. The assumption is that viewers will be identified in an network specific manner and that requests from a set-top over the RC will be from a known viewer identity to the relevant server that can handle the request.

6.3.2 Transactions and State Handling

Communication between server and client components of enhancements might involve numerous return channel transactions and enhancement state changes.

The client side enhancement components are required to track the state of the application. *Federated* transaction handling (transaction handling across multiple servers) is not in the scope of this section.

The server side enhancement components provide support and data sources for applications but do not maintain the state of the enhancement itself.

Enhancements should be designed to behave properly if the headend servers they depend on are not accessible for some reason. Precise behavior is specific to each enhancement.

Applications should recognize the following states and transitions:

- Starts out disconnected
- Starts out connected
- Transitions to connected from disconnected
- Transitions from disconnected to connected

6.3.3 Network Addressing and Routing

The addressing of set-top devices and the routing of their RC requests is handled by the infrastructure in place in the headend. This might include technologies such as DSG, DOCSIS, legacy OOB and other combinations. The link between a set-top and the termination point within a cable network is maintained over the HFC network.

The contact point between the HFC network and external networks is defined as the ***proxy server*** (see diagram below) and will be seen as the source of all RC traffic requests from all set-top clients on the HFC network.

The following diagram shows a slightly more detailed view of the headend configuration. Also see the more detailed proposed architectures below this.

Note that the dashed arrow transactions on the left show a *virtual* transaction between the set-top application and an application server. The optional EBIF translation step using an EBIF RC server is shown as 1 followed by 2b and 3 while the direct transaction is shown as 1 and 2a where the application server does the EBIF response formatting itself.

Actual data flows would be passed through the *headend equipment* shown in the gray cloud.

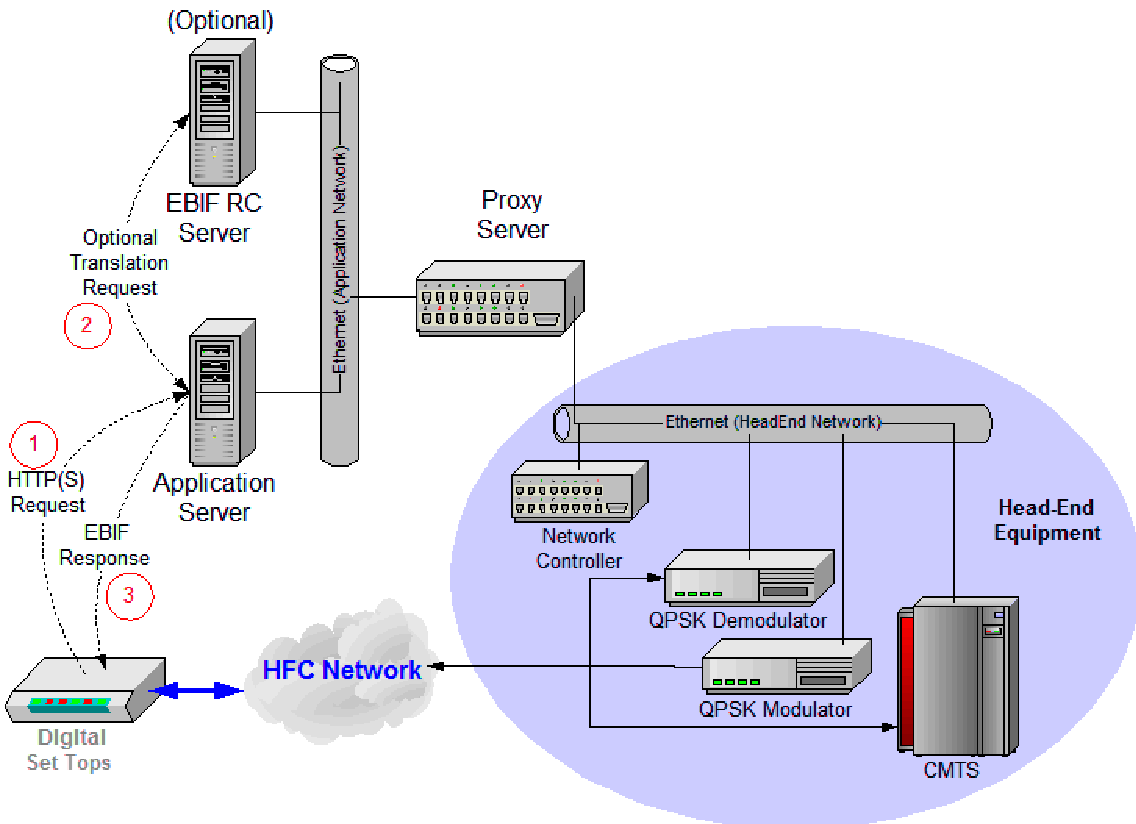

Figure 6–1 - Networks at the headend

6.3.4 Data Format Verification

Data files accessed between client and server components of applications should be verified to insure the validity of the data before they are exchanged.

This document does not describe the verification/validation process.

6.3.5 Application and Data Source Authentication

Network operators or programmers may require that applications and their data sources be authenticated. This document does not describe how this may be accomplished.

6.3.6 URI and Message Formats

These are as described in [ETV-BIF].

6.3.7 Low Level Transport Handling

The following low-level transport details are addressed by the particular network protocol, such as legacy OOB or DSG, in use in a given situation.

- Connection set-up and breakdown
- Session handling
- Addressing and routing
- Message delivery, acknowledgement and retries
- Error generation and propagation
- Latency and Quality of Service
- Return channel memory allocation
- Allocation of return channel resources to enhancements
- Other low-level issues

6.4 Architecture

The following architecture illustrates one possible scenario showing the client set-top box communicating with the headend and then having the requests forwarded to application servers also hosted in the MSO network.

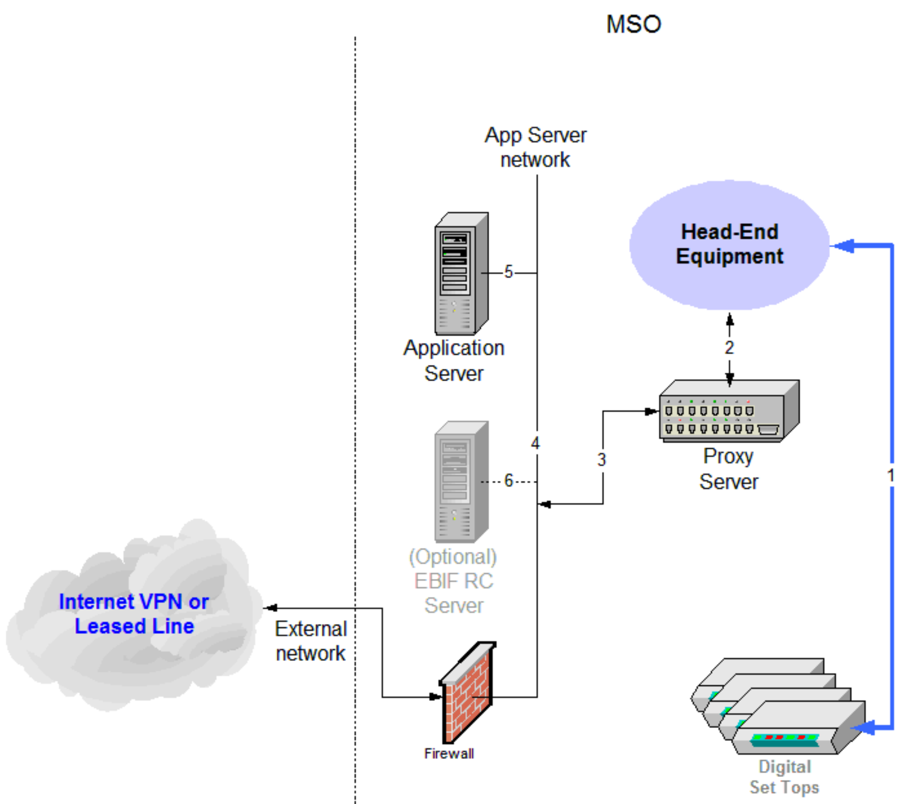

Figure 6–2 - MSO hosted application server network

A firewall may exist between the headend and the proxy server (link 2) or between the proxy server and application server network (link 3 or 4).

The assumption is that the EBIF RC server operates at a lower level in the OSI stack. It therefore can be called a *gateway*. It will also be housed in the MSO to perform protocol transformation only. It converts from external data formats to EBIF. This does not include addressing translation.

6.5 Interface Matrix

The following interfaces are shown in Figure 6–2 above. All are bi-directional. Numbers correspond to the numbered links on the figure.

Table 6–1 - Description of Interfaces in Figure 6–2

No.	Description	Protocol	Security
1	Set-top to headend	Proprietary or standard-based (not in scope)	Handled by infrastructure (not in scope)
2	Headend route to proxy server	HTTP or HTTPS from headend to network, EBIF binary from network to headend	Unencrypted, SSL, TLS, fire-walled
3,4,5,6	Internal MSO data network (possibly with subnets)	Request (HTTP, HTTPS, RTSP, other internet protocols). Response is EBIF binary.	Unencrypted, SSL, TLS, fire-walled

The next diagram shows the layout with the broadcaster housing the EBIF RC server and application server.

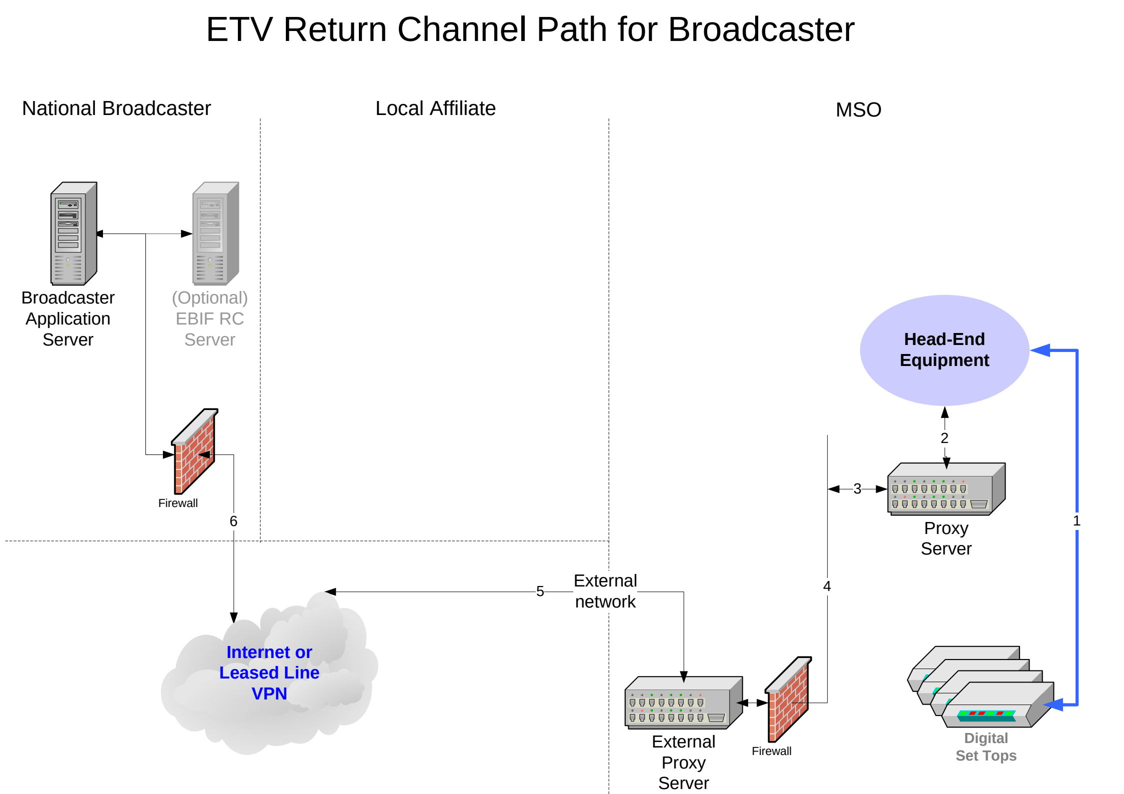

Figure 6–3 - Broadcaster hosted application server

6.6 Channel and Content

The data channel is provided by the headend infrastructure and connects the set-top to the proxy server to the application servers. The EBIF RC can then optionally provide translation to EBIF binary format for responses and route them back to the headend.

The content will be provided by the various servers as described in the list under **Server Roles** above. Content can be returned in a number of formats and is converted to EBIF binary format by the EBIF RC server.

The proxy server is responsible for message routing to the proper application servers only. Identification of the correct application server is based on information encapsulated in the URI request identifies the application owner (MSO/Local Affiliate/National Broadcaster).

The set-top box initiates an interactive session contacting the application server by IP using HTTP or HTTPS v1.1 POST requests. For these purposes only application and EBIF RC server IP addresses should be registered in the Network Controller database.

To contact the application server the set-top box should know the IP/Hostname and Port in advance. It may be either *hard-coded* in the user agent or injected in the EBIF data stream by the local MSO EBIF Streamer.

This information is assumed to be available to the set-top box. No details about this implementation will be described in this document. This will be dependant on the network implementation and will apply to legacy architectures and also future technologies such as OCAP.

6.7 Transaction Types

The following four types of transactions have been identified. The following table describes the properties of the transactions.

The following columns are used:

Timing: Describes the response time needed by the system in processing this transaction with real-time meaning as instantaneous as possible, near real-time meaning within an acceptable delay period and batch meaning at a later date.

Auditable: Describes the requirement for logging of a transaction for very specific time periods. The period depends on legalities surrounding the transaction itself.

Droppable: Describes how critical each transaction is specifically.

Acknowledgement: Describes how the application is expecting a response.

Table 6–2 - Transaction Type Attributes

Type	Timing	Auditable	Droppable	Acknowledgement	Example
Timed critical	Real-time	Yes	No	Yes	Calculate cost for a given service in up-sell enhancement
Timed non-critical	Real-time	No	Yes, winner determined by statistical analysis	Yes	Music show vote
Commerce transaction	Batch	Yes	No	No, application can provide feedback as in "Transaction processed"	Purchase of an item or service from the TV that can be handled in a batch mode
Timed transaction	Real-time	Yes	No	Yes	VOD purchase, ring-tone purchase

6.8 Application Server Load Handling

This section describes options available for handling return channel load at the application server level and *not* at the interactive network. Options available here include:

- Hardware load balancers
- Dedicated front-end servers (such as the authentication servers)
- Distributed application and database servers
- Throttling of incoming traffic
- Redundant hardware

6.9 Loading Guidelines

This section describes some real-world return channel load guidelines for various shows and enhancements.

A deployable voting/polling application should be scalable up to 1million votes, with user responses expected within 15 seconds of the voting request.

For a system of this size, a solution may make use of up to 100,000 simultaneous direct TCP socket connections, utilizing very small return messages (<50 bytes) sent to a set of load balancers. The load balancers then forward the messages to the consolidating network.

To further alleviate flooding of the return channel, clients may stagger their responses by a random interval between 0 and N seconds, with 5 seconds probably the upper limit of this window.

From a business perspective, asking users to pay per vote might help defray some of the return channel infrastructure costs.

6.10 Use Cases

The following use cases show different application scenarios and the data flow through the system. They are arranged in order of increasing complexity. Please note these architectures are representative and may be more complex or different depending on the environment at the MSO or National Broadcasters.

6.10.1 MSO Only

This scenario is where the MSO is running applications only on a national basis and there are no other ETV applications with RC. Content is provided by content servers hosted and managed by the MSO themselves.

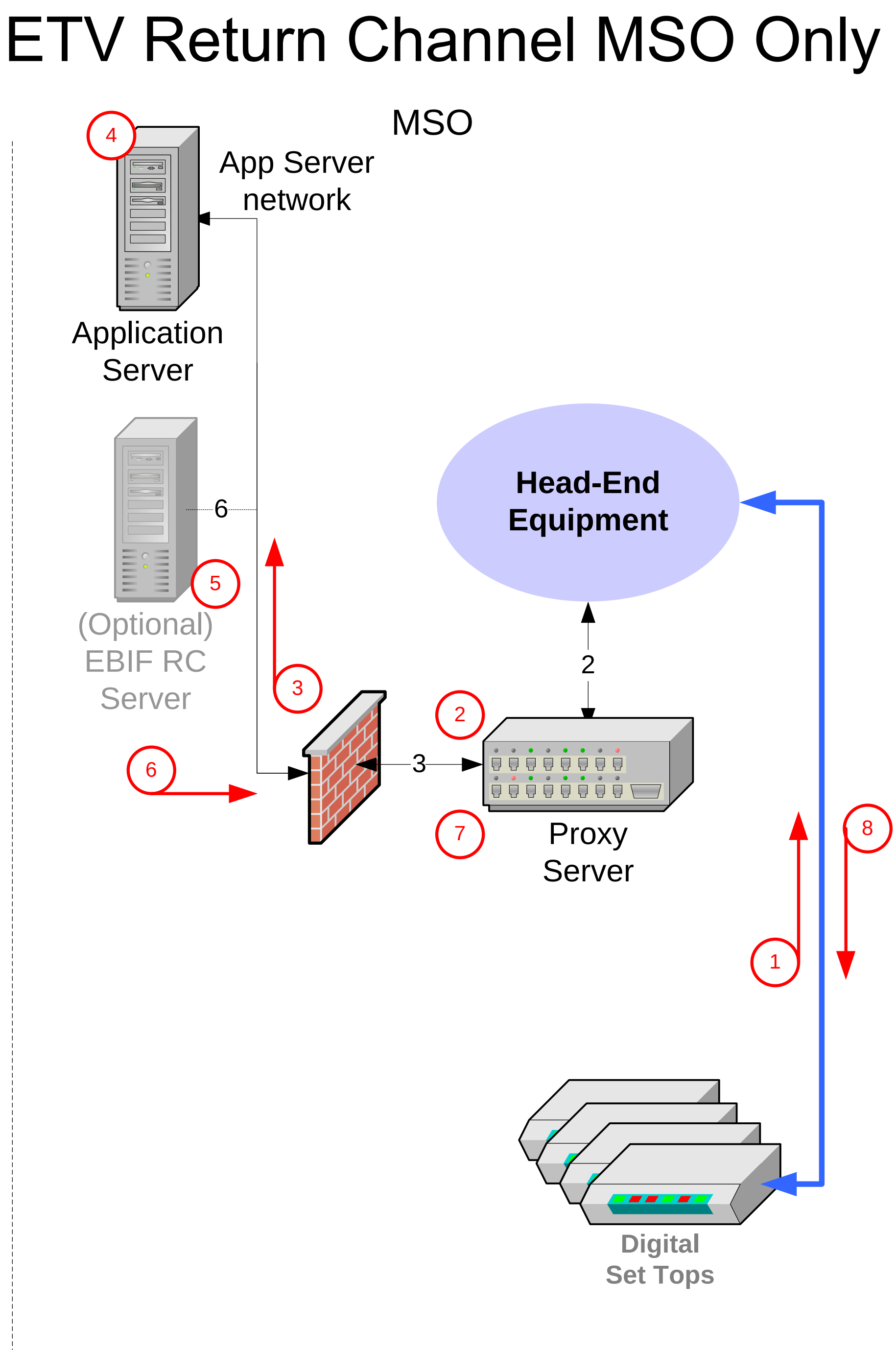

Figure 6–4 - MSO Only

Notes (numbers correspond to red circled numbers above)

1. Set-top box issues request over headend infrastructure (protocol and addressing details not shown).
2. Headend and proxy server resolve addressing and route request to application server as requested in the EBIF URI POST request.

3. Request is routed internally in MSO network to Application Server network.
4. The application server processes the request.
5. (Optional) The HTTP(S) response is routed back to the EBIF RC server that converts the response to EBIF data formatted for the set-top. Alternatively, the application server itself may produce the response in EBIF format directly (in this case, skip this step).
6. EBIF response is routed to proxy server.
7. Proxy server maps addresses and routes response to headend destined for the original set-top box.
8. The headend infrastructure routes the response back to the originating set-top.

6.10.2 Broadcaster Only

This scenario is where the Broadcaster is running applications only on a national basis. Content is provided by application servers hosted and managed by the Broadcaster themselves who also can host the EBIF RC server.

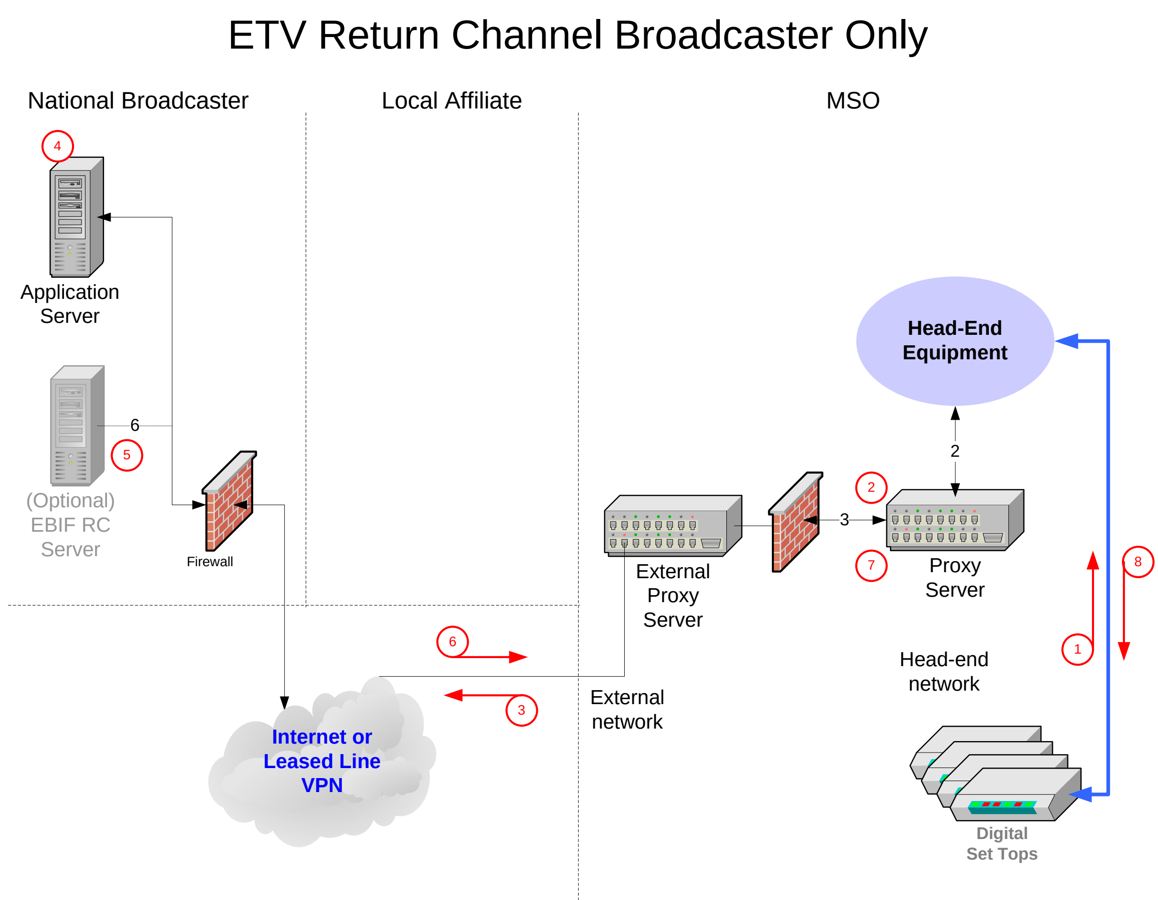

Figure 6–5 - Broadcaster Only

Notes (numbers correspond to red circled numbers above)

1. Set-top box issues request over headend infrastructure (protocol and addressing details not shown).
2. Headend and proxy server resolve addressing and route request to application server as requested in the EBIF URI request.

3. Network routes request outside MSO network via an external proxy server using HTTP or a dedicated socket connection.
4. The national broadcaster application server processes the request.
5. (Optional) The HTTP(S) response is routed back to the EBIF RC server that converts the response to EBIF data formatted for the set-top. Alternatively, the application server itself may produce the response in EBIF format directly (in this case, skip this step).
6. The response is routed back to the MSO via the data network using HTTP or a dedicated socket connection.
7. Proxy server maps addresses and routes response to headend destined for the original set-top box.
8. The headend infrastructure routes the response back to the originating set-top.

6.10.3 MSO and Broadcaster Combined

This scenario is where the MSO and Broadcaster are running applications simultaneously and the applications with RC are connecting to servers at both the MSO and broadcaster. Both the MSO and broadcaster hosting an EBIF RC server for conversion purposes are shown, although this server is optional.

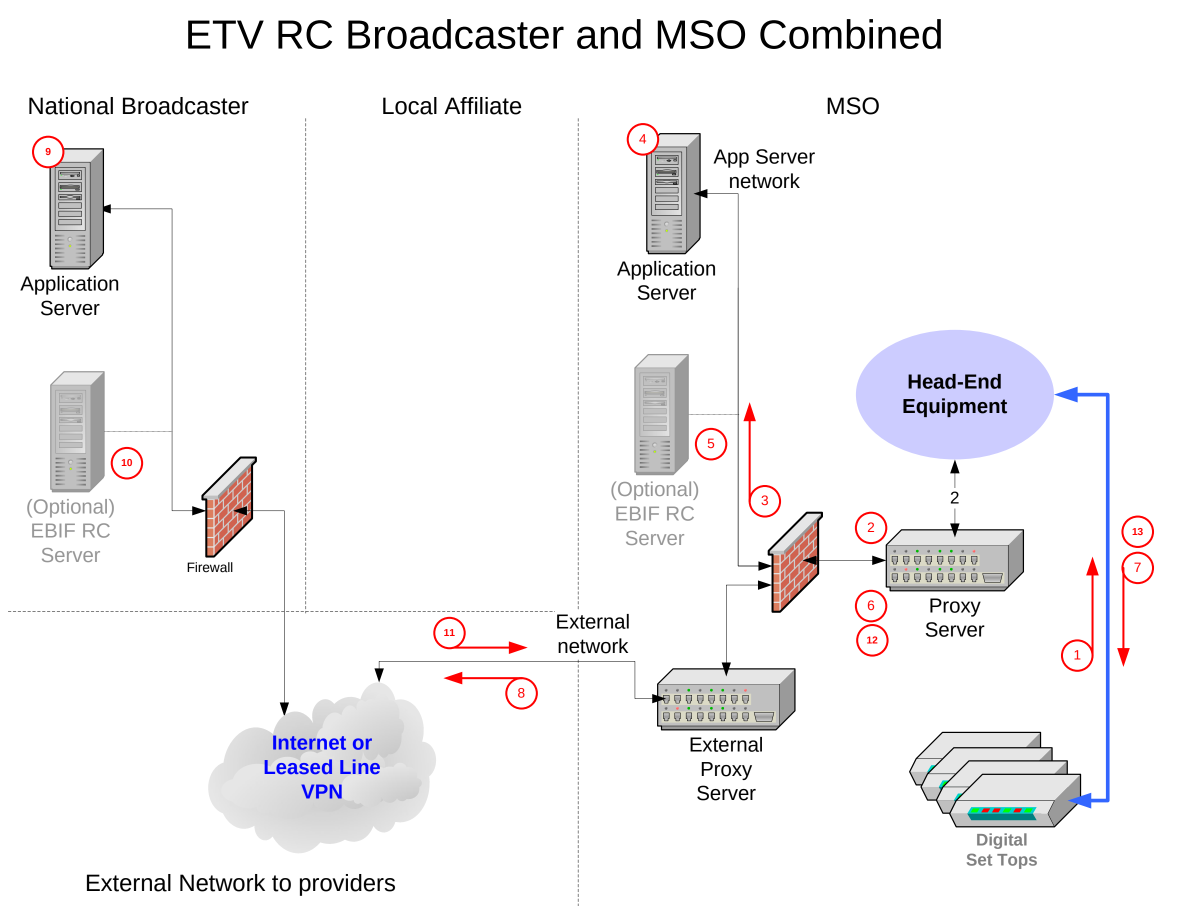

Figure 6–6 - Broadcaster and MSO combined

Notes (numbers correspond to red circled numbers above)

1. Firstly the set-top box issues request over headend infrastructure (protocol and addressing details not shown) targeting a server in the *MSO application server network.*

2. Headend and proxy server resolve addressing and route request to MSO's application server as requested in the EBIF URI POST request.
3. Request is routed internally in MSO network to application server network.
4. The application server processes the request.
5. (Optional) The HTTP(S) response is routed back to the MSO's EBIF RC server that converts the response to EBIF data formatted for the set-top. Alternatively, the MSO's application server itself may produce the response in EBIF format directly (in this case, skip this step).
6. Response is routed to proxy server.
7. Network controller maps addresses and routes response to headend destined for the original set-top box.
8. Now consider if the application server targeted is outside the MSO network and resides in the *National Broadcaster network*. Steps 1 and 2 are similar as above. Now see point 8 where the data request is routed via the external proxy server to the remote network using HTTP or a dedicated socket connection.
9. The national broadcaster's application server processes the request.
10. (Optional) The HTTP(S) response is routed back to the national broadcaster's EBIF RC server that converts the response to EBIF data formatted for the set-top. Alternatively, the national broadcaster's application server itself may produce the response in EBIF format directly (in this case, skip this step).
11. The response is routed back to the MSO via the data network using HTTP or a dedicated socket connection.
12. Proxy server maps addresses and routes response to headend destined for the original set-top box.
13. The headend infrastructure routes the response back to the originating set-top.

7 ON DEMAND CONTENT

While today's On Demand assets are accessed primarily via menus within a program guide or similar application, ETV applications present the opportunity for a more integrated and targeted user experience. Rather than viewers browsing or searching for On Demand content, the synchronous nature of ETV allows applications to present viewers with content related to the programming they have already selected. This may be used to enhance the main programming, extend the reach of advertisers, or raise subscriber awareness of VOD.

The challenge lies in coordinating between the broadcast and On Demand systems. Historically, these have been separate services with distinct business systems, advertising models, delivery networks and user interfaces. Although the requirements for applications and the EBIF specification are relatively small, there are considerable system and operational requirements. The following sections define the possible types of applications and content, followed by more detailed deployment scenarios.

7.1 Content Sources

The term "On Demand" generally refers to any content that can be delivered to an individual user whenever they want it, versus broadcast content that is sent to all subscribers on a fixed schedule. Possible sources for On Demand content include

- Video On Demand (VOD) servers
- Digital Video Recorders (DVR)
- Streaming media received through future IP tuners

Ideally, the source and physical location of the content should be transparent to the application, with a logical application reference mapping to potentially different sources depending on the local client and network capabilities. However, each delivery method has unique implementation and deployment requirements, and early solutions will likely be much more limited in scope.

Because standards do not yet exist for delivering assets to a DVR, and because ETV's initial goal is to enable interoperability across all deployed set-top boxes, most of which do not support DVR or IP tuners, this discussion focuses exclusively on VOD delivery.

7.2 Applications for On Demand Content

Two broad categories of applications are defined below along with common examples.

7.2.1 Long Form Content

Long form content is the most common and straightforward integration between ETV and On Demand. It is designed to increase the value of advertising or programming by linking between linear and related On Demand content. Common examples include:

> **Telescoping Ads**—Combines a standard in-program 30-second commercial with an On Demand long-form advertisement, typically three to twelve minutes long. Viewers are given a call to action during the 30-second ad, allowing them to click for more information or advertainment content.
>
> This same concept could be applied to network promotional advertisements. For example, the application could telescope to a season premiere sneak preview video from a promo ad.

Bonus Material—Allows broadcast programs to include extra content similar to DVDs. For example, the ETV application for Lost might include a menu with short, behind the scenes videos or past episode highlights. This uses the same mechanism as telescoping ads, but will likely require different business rules.

Bookmarks—Applications could include references to On Demand content that the user adds to their "saved" list, allowing them to view the content at their leisure. This is interesting as it avoids interrupting the viewing experience, and helps scatter the simultaneous demands on VOD systems.

7.2.2 Indirect References

In addition to direct references to long form video assets, an application may wish to make indirect references to content stored on a VOD server. This might include references to a "folder" or sub-menu within the program guide's VOD application or a storefront virtual channel. It is also conceivable that an EBIF application could be stored on a VOD server.

Examples of indirect references include:

FOD Promotions—ETV applications on a family of channels could include a link to their FOD menu. This would be a fairly static link that isn't tied to a particular asset, but instead to their branded sub-menu.

SVOD Promotions—A premium channel's ETV application for an original series could link to their SVOD menu listing past episodes. Depending on the user agent or the guide/VOD application capabilities, this might include an upsell feature for non-SVOD subscribers.

Pay VOD—A 30-second network ad for the theatrical release of a movie sequel could prompt the user to view the original movie on VOD. If the user clicked on the prompt, they would be taken to the VOD purchase screen for that asset.

Virtual Store Fronts—A 30-second spot for a car could link to the local car dealer's sub-menu within the MSO's advertising on demand application (e.g., FreeZone).

The current EBIF specification only allows VOD locators to be referenced within page and video widgets and does not yet have constructs for indirect references. ***Therefore, indirect references are currently outside the scope of this guideline document.***

However, it is possible for some of these applications, such as virtual store fronts, to be implemented as EBIF applications with direct VOD references. Thus, the 30-second spot could launch an EBIF storefront application that contains multiple direct VOD references rather than linking to an MSO's advertising menus within the VOD application. It is also noteworthy that existing On Demand sub-menus can be accessed as virtual channels by EBIF applications using channel tunes, where this capability has been made available by the MSO. This might be useful for local applications that depend on the local channel line-up, such as a local enhanced ad.

7.3 Long Form Content Scenarios

This section discusses direct references to VOD assets from within an EBIF application. Figure 7–1 illustrates the typical workflow for an application using long form content. The example walks through the flow for asset distribution, application broadcast, and application execution. The application provider will create an application referencing a VOD asset, and source the VOD content for distribution to participating cable systems in advance of the application broadcast. The cable systems will receive and propagate the asset throughout their network. The ETV application is later broadcast along with the scheduled programming, and is forwarded by the cable system. Before launching the application, the system validates its permissions and verifies that the asset is available on the local system. Once the viewer requests the VOD content, a session is established with the VOD server and the asset is played.

The diagram includes conditional error handling based on the type of application. VOD dependent applications, such as telescoping ads, use VOD content as an integral part of the ETV experience and should not be displayed if the content is unavailable. VOD-capable applications, on the other hand, use VOD content as a minor element and could continue to operate with a subset of features. For example, an enhanced reality show with VOD "bonus material" could just disable the VOD related menus.

Operations within the "ETV Infrastructure" box might be done by headend components, the User Agent or ETV applications depending on the selected implementation.

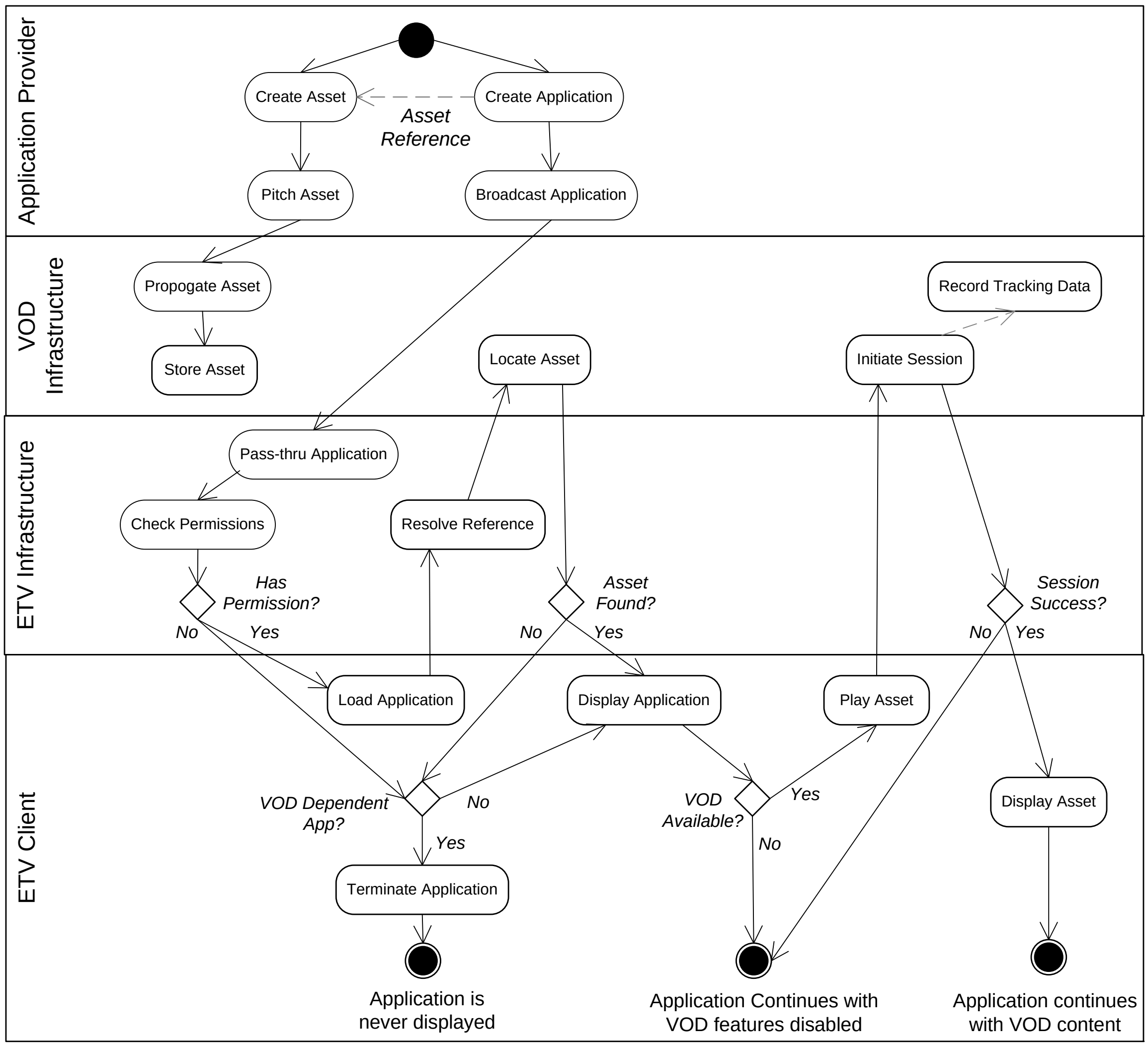

Figure 7–1 - On-demand long form content scenarios

The required processing can be grouped into four general areas: asset reference and propagation, reference validation and resolution, presentation, and reporting. The next sections list potential solutions and the pros and cons of each.

7.3.1 Asset Reference and Propagation

In order for an EBIF application to request long form content, a unique identifier must be associated with each VOD asset and this identifier must be preserved as the asset is propagated from the content creator across all the cable systems' VOD servers. Asset references must support hierarchical identifiers so that application providers can manage their own asset IDs without needing to coordinate with other providers. The CableLabs Asset Distribution Interface [ADI 1.1] defines the Provider_ID and Asset_ID metadata fields for this purpose. Provider_IDs are centrally managed to ensure they are unique, and each provider is then responsible for managing asset IDs within their domain.

Although the return channel network is used to communicate with the VOD servers, EBIF applications will not use the forms mechanism described in Return Channel, Section 6.1 above, to request VOD assets. Instead, VOD assets are treated as another type of video source, in addition to MPEG services, that can be referenced within page and video widgets. VOD references use resource locators with a Video On Demand URI Locator Scheme containing the provider ID and asset ID as described in the EBIF specification.

An application provider must also properly encode and deliver VOD assets to each participating system operator. The CableLabs' Content and Encoding specifications and ADI compliant pitcher, catcher, and asset management systems available today should provide the required infrastructure for asset delivery. However, these systems were designed primarily for Movies on Demand (MOD) applications, and extensions may eventually be necessary to provide the desired management interfaces. For example, advertising business models may require different metrics than those used by MOD.

Figure 7–2 depicts a typical delivery architecture. The VOD hierarchy within a cable system is network dependent. The EBIF application delivery path could be any of the scenarios described in the Cable Network Distribution sections.

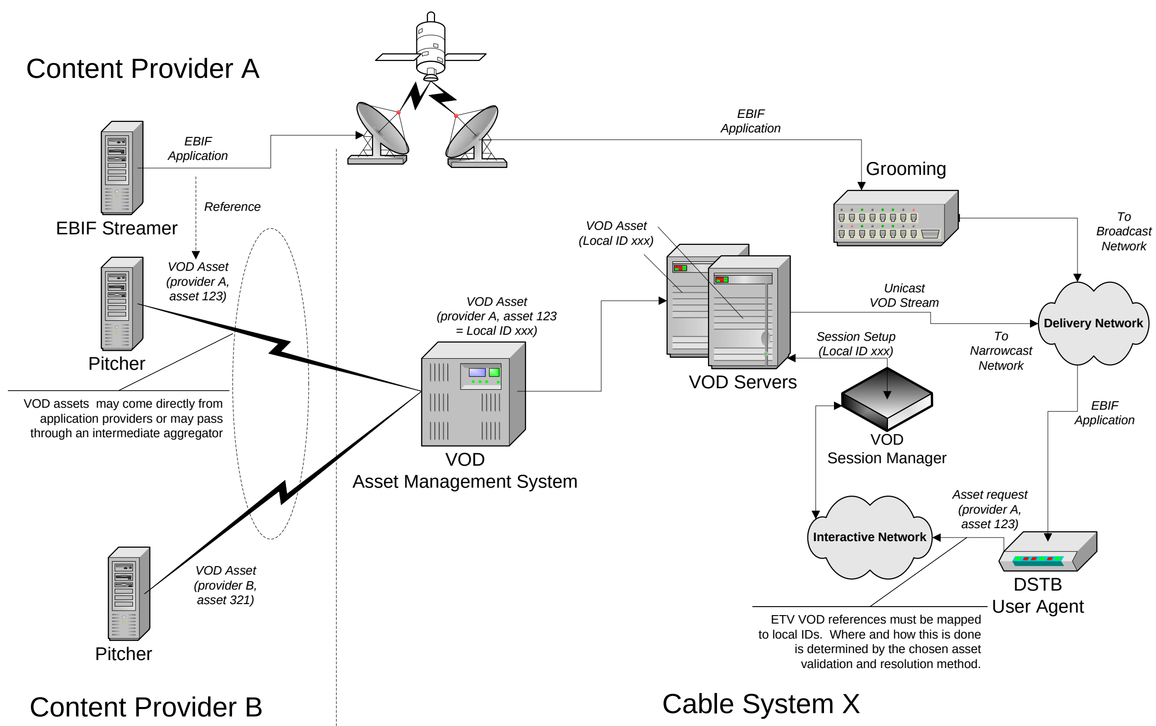

Figure 7–2 - Content Referencing and Propagation

Known Issues:

- Asset URIs may need to be mapped to local identifiers.

  The ADI Provider_ID and Asset_ID fields may not be maintained all the way through the system. VOD metadata processing has evolved over time and varies between systems and servers. For example, not all deployed systems have upgraded to ADI 1.1. Also, because deployed VOD applications browse a local asset catalog, asset IDs are often treated as private data and the ADI identifiers may be mapped to internal IDs during propagation. Thus, the ID used in the session setup request is not the same as the ID used by the pitcher/catcher and asset management systems.

It is also possible that a logical EBIF reference could map to different physical assets to allow future decision support engines to resolve references based on local business rules. For example, a telescoping ad might have a generic reference that is mapped to different local or targeted VOD assets during resolution.

- Propagation delays may vary between systems. Because it can take days for an asset to propagate all the way to an edge VOD server, it may be necessary to build average latency windows into business processes. For example, if the average propagation delay is two days, ETV providers cannot expect to refresh VOD content any faster than this.

7.3.2 Reference Validation and Resolution

Reference validation ensures that the VOD content for nationally broadcast applications is locally available before prompting the viewer. This is necessary to provide an acceptable user experience as it is highly undesirable for the application to offer content that is unavailable. Early VOD experiences found that if viewers receive errors on their initial attempts, they are not likely to try and use the feature again in the future.

An application may not be able to access long form content on all systems depending on business agreements and network capabilities. Also, VOD systems use private networks and protocols, each with varying propagation delays, so implementations must verify if the asset is physically available on a local server and map the generic reference to a network specific location.

There are three reasons why an asset may not be available:

1. The application is not authorized on the local system. Some operators will choose not to carry a national application on some or all of their systems. In this case, the application should never be launched.
2. The application is not authorized to access VOD on the local system. This may be the result of business agreements, resource limitations, or because the system does not support VOD. For VOD-dependent applications, the application should not be launched. VOD-capable applications should be launched, but should disable VOD related features.
3. There was an error delivering the asset to the local system. This should be handled the same as case 2 for VOD dependent and VOD capable applications.

The first case applies to all applications and the second to any application requiring limited resources such as VOD or the return channel. Therefore, these are not VOD-specific issues and are addressed in the general security/permissions discussion. ***Once permissions have been validated, the required assets should be available on the local system, and asset validation centers on the third case: verifying that the asset physically exists to ensure a robust user experience in the event of propagation errors***.

Reference resolution maps the generic ETV identifier to an identifier recognized by the local VOD system. The mapping should be transparent to the application, and the local system will manage this either as part of asset validation or in real-time during session setup. The specifics of how this is done and how the User Agent or VOD system locates the server containing the necessary content are implementation dependent and will vary based on the network and VOD system.

Content synchronous applications present unique design issues because they often have large viewing audiences and could overload the network with a flood of simultaneous requests. VOD-dependent applications are particularly time sensitive as they must verify references during initialization before displaying anything to the user. For example, a 30-second telescoping ad during the Super Bowl could create a huge response spike if every application queried for asset availability in real-time.

In order to better understand the system implications, potential solutions are listed below.

7.3.2.1 Human Monitoring

Albeit inelegant, systems could use manual processes to detect propagation errors. Early deployments are likely to be fairly static with operators manually configuring catcher systems and grooming equipment to accept ETV content. Therefore, operators would be aware of what content is required and could monitor the local systems to ensure assets were successfully received and propagated. In the event that the required assets are unavailable, system operators could filter applications either using the grooming interfaces directly, or by disabling the permissions for an application.

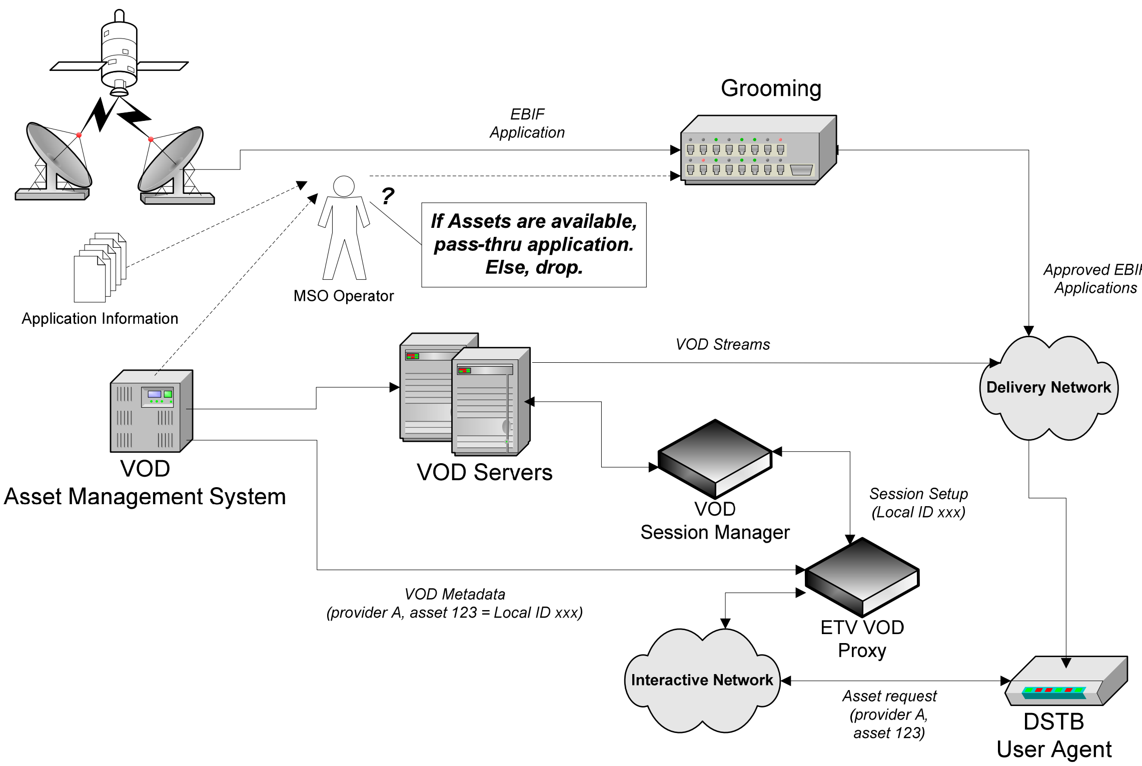

Figure 7–3 - Human Monitoring

Implementation Requirements:

- Manual process for operators to cross check asset availability. Given the operational overhead, this is best suited to systems using more centralized VOD architectures.
- Headend proxy servers to map the logical ETV reference to a local asset ID during session setup.
- No client side modifications are required as applications would be filtered at the headend.

Known Issues:

- This solution assumes that a robust permissions mechanism is in place and that the reference validation step is serving only as an additional layer of error checking.
- Because this is a simple on-off mechanism, it would not allow VOD-capable applications to disable features based on asset availability.

7.3.2.2 Headend Filtering

Headend filtering processes metadata broadcast along with the application to determine if the required assets are available on the local system. In the event that the assets are not available, the application could be dropped from the multiplex or the signaling could be updated to notify the client.

Implementation Requirements:

- Extensions to the application signaling to enable real-time grooming equipment to determine the set of required assets. In order to allow time for the necessary processing, the signaling may need some form of "pre-load" mechanism that signals the need for an asset in advance of the application.
- A grooming mechanism integrated with the VOD system to process the ETV broadcast stream, validate that the assets are available, and drop the applications or update the signaling as necessary.
- Headend proxy servers to map the logical ETV reference to a local asset ID during session setup. Alternatively, the signaling could be updated with a local asset ID and reinjected into the broadcast.
- No client side modifications are required as applications would be filtered at the headend.

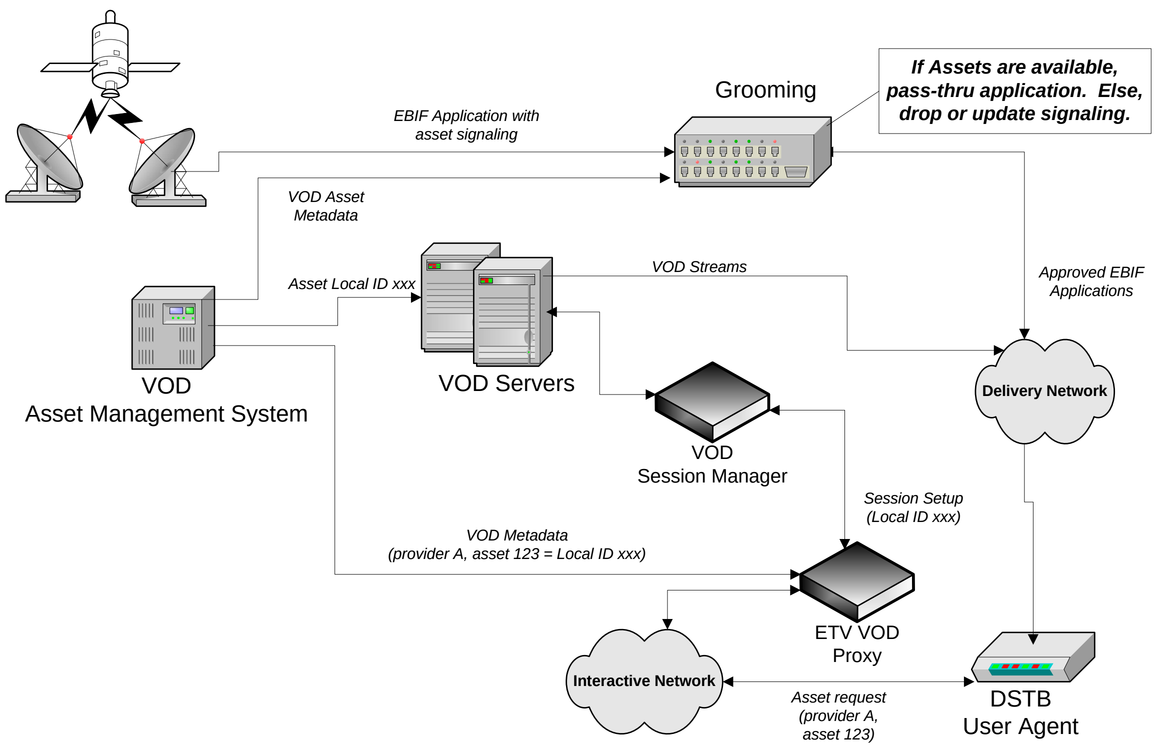

Figure 7–4 - Headend Filtering with Proxy

Known Issues

- This would require new grooming functionality and integration between the grooming equipment and VOD asset management systems in every headend.
- The broadcast topology generally does not match the VOD server topology, and the last point of decision for application grooming may span multiple VOD servers. Thus, grooming based on physical asset availability could be inefficient and difficult to implement.
- In order to support VOD-capable applications, the grooming would need to update and reinject the signaling to indicate the availability of each asset rather than just filtering the entire application.

Given the above issues, headend filtering is not recommended for determining if the asset is physically available. However, it might be a viable option for validating if the application has VOD permissions. That level of filtering would be based on general policy management and would not be dependent on the VOD topology or require integration with the VOD system.

7.3.2.3 Client Validation

Instead of filtering applications and/or their signaling in the headend, the client could perform the asset validation. In these scenarios, either the User Agent could initiate the check before launching the application, or the application could explicitly query for availability.

It is important to note that any client-side solution must consider the processing, memory, and code size limitations on base profile devices.

7.3.2.3.1 Server Queries

This scenario would send interactive queries to the VOD system in order to validate each asset reference. While return channel and server capacity varies between systems, the synchronous nature of ETV applications could easily exceed the simultaneous capacity of deployed systems. The worst case example of a VOD-dependent 30-second ad within a popular program would require every client to query the server within a few seconds window before launching the application. While this may be an option for future DOCSIS-based return channels, it is problematic for legacy networks.

7.3.2.3.2 Local Validation

In order to minimize network traffic, the client would validate the reference using information available on the set-top, similar to the catalog solutions used by Movies on Demand applications. The catalog data could also be used to map between the ETV reference and the local asset ID.

Several methods exist for delivering this data either via broadcast or OOB channels, each with their own trade-offs.

- Local asset catalogs can be broadcast in-band on each transport so that the client can access the data without tuning away. In order to optimize bandwidth usage as the number of ETV VOD assets increases, the catalog generator could be aware of the programming schedule and only include assets that are required by active applications. This solution implies a fair amount of broadcast overhead and requires the catalog generator to determine which assets are used at which time.
- The catalog can be continually spooled over the out of band, eliminating the need to broadcast it on every transport. However, OOB bandwidth is very limited, and catalog size would have to be highly optimized as in case 1 above. Bandwidth requirements must also be weighed against any other two-way applications the operator may have deployed.
- The catalog can be a proprietary extension to the deployed MOD catalog or guide data. This could work for solutions that cache the data on the client. But, this will not work for memory limited boxes which typically tune to a different transport or request data in real-time via the OOB.

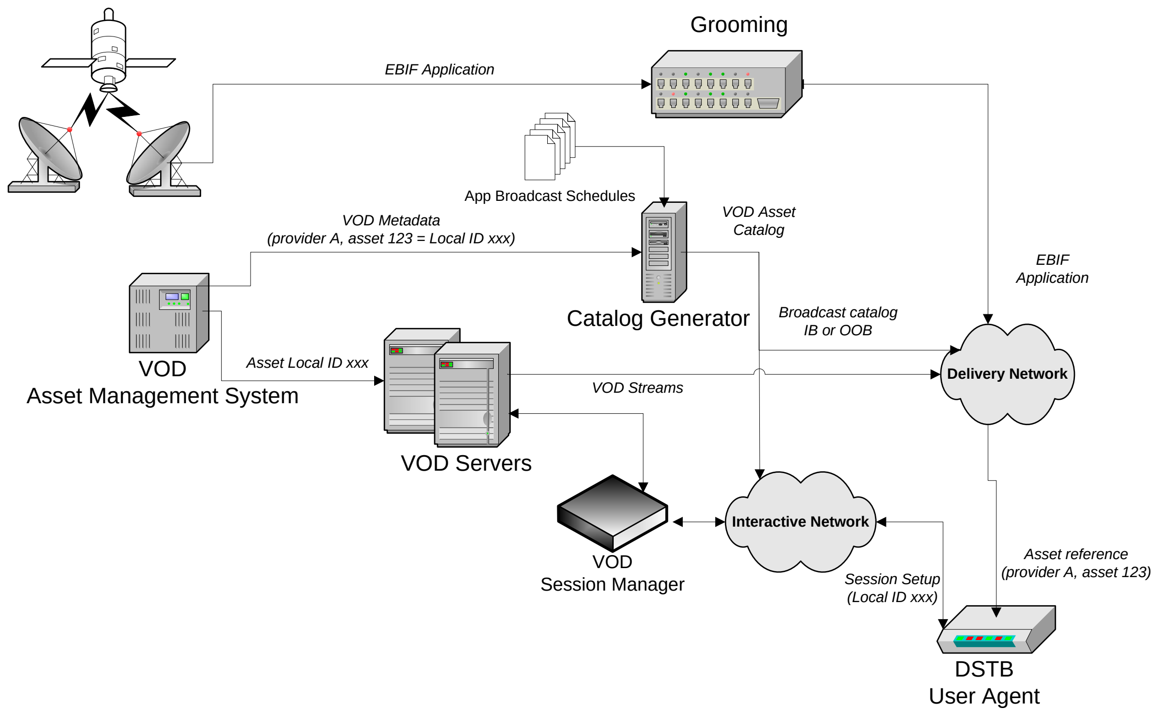

Figure 7–5 - Catalog Based Validation and Resolution

Implementation Requirements:

- Catalog generator optimized for the local network and client.
- EBIF interface for applications to determine if assets are available. This could either be an explicit query or the User Agent could automatically update the resource table when loading the application. How the User Agent validates and resolves the reference is implementation dependent, and could just be a pass-thru to the underlying VOD implementation.

Known Issues:

- Adds additional processing on the User Agent and/or application.
- May not work on all networks or clients because of insufficient bandwidth and/or memory.

7.3.3 Content Presentation

Content presentation has minimal system implications. Applications will follow the methods defined in the EBIF and User Interface Guidelines, including how to display the content on the screen using references to VOD URI locators within page and video widgets and how to manage trick mode using Video Control Actions. The details of how a session is established and controlled are proprietary to the VOD system and User Agent, and are hidden from the application. It is assumed that an EBIF User Agent will use the deployed VOD protocols and no new interfaces nor features will be required.

Because of the potential for synchronous applications to overload the network or VOD servers with simultaneous requests, implementations may need to consider algorithms to randomize session setup requests. Applications can also be designed to spread out VOD usage. For example, rather than tuning directly from an in-spot prompt to a VOD session, the application could load a main screen with text/graphical information and menus for different VOD assets. Subscribers would then spend varying amounts of time browsing this screen before launching a VOD session.

8 DVR CONTROL

DVRs can operate as both a source for On Demand content as discussed in the previous section, as well as a timeshifting/recording device. DVR control focuses on the latter case, and example applications might include:

- Pause live TV in order to link to interactive app or telescoping ad;
- Schedule a remind/record from an EBIF application within a promo of an upcoming broadcast program.

These features will require standard EBIF APIs, but the implementation will likely be tightly linked with the User Agent and DVR / Guide application. The primary system implications surround metadata with similar asset reference and validation issues as for on demand.

For example, the system must be able to reliably link a unique identifier in an application with the program in the broadcast schedule in order to schedule a remind/record. This could require integration with the guide database either in the headend or in the client, but some legacy set-top boxes only retain a small number of days of EPG listings. Therefore, if the check is set-top box based, then the promotion must account for this. Also, if a program is not available on the cable plant, presenting the subscriber with an offer to record just creates subscriber dissatisfaction.

9 CAPACITY MODELING

Because ETV applications are synchronized with the program broadcast, they have the potential to generate extremely high simultaneous subscriber participation. This requires careful design for applications that use limited network resources such as the return channel, VOD sessions, or response servers. To use VOD as an example, most deployed VOD systems assume 10% simultaneous usage from the digital subscribers within in a hub. And, even though traditional VOD usage has peak hours such as Friday night, subscribers browse VOD catalogs at their own pace and thus session setup requests are random. In a synchronous ETV application, however, all subscribers are prompted to view content within a very narrow time window.

Modeling should account for:

- Projected requirements for interactive network bandwidth;
- Projected requirements for session resource managers (legacy network controllers and VOD session managers);
- Projected requirements for application servers (VOD servers, response servers, etc.).

Calculations should estimate the number of simultaneous viewers based on the average number of actively viewed channels and the average number of channels with enhancements at any given time.

10 SECURITY

10.1 Background

Unlike OCAP applications that are limited to running on advanced <Tru2way> Hosts, eTV EBIF applications can run on most legacy set-top boxes (STBs), as well as on the advanced devices. As these applications become prevalent in more markets, developers will want to build applications that have advanced capabilities. Unlike OCAP applications, eTV applications do not currently have a well-defined security framework. This was because the security requirements for initial set of use cases supported on the eTV platform (like a voting application) were relatively low.

This limits the scope of services that can be provided to customers via eTV. A security framework and infrastructure is being developed so that more powerful applications can support advanced use cases and handle sensitive customer information, such as credit card numbers.

The rest of the security section is under development.

10.2 Use Cases

This section describes a few use cases that have security implications. The initial use cases that require only limited security support are listed below:

1. Voting applications: Subscriber can vote for their favorite contestants or players.
2. Telescoping advertisements: Subscriber may click on a button to view a longer streamed advertisement. Subscriber may also request the VOD ad to be bookmarked so that they can view it later over On Demand.
3. Request for Information: Subscriber may request more information to be mailed or emailed to them. Subscriber may also request a phone call.

The security architecture and infrastructure that is under development for advanced eTV applications need to consider additional use cases, a few examples are listed here:

1. Impulse Pay Per View and Upsell: Subscriber orders TV programming or content while watching a channel. Payment for the order is charged to subscriber’s monthly bill. The subscriber does not submit any personally identifiable information during this transaction.
2. Video shopping channels: Subscriber purchases items from vendors (like QVC or HSN) using a previously established account which is maintained by the vendor. Subscriber’s payment and shipping information, address and phone number have been previously configured with the vendor. Some combination of subscriber phone #, zip code and/or a passcode are entered on the TV when actually completing an order.
3. Impulse purchases: Subscriber orders advertized items associated with the program that they are viewing, like baseball paraphernalia advertised while viewing a game. The transaction would be fulfilled by a vendor on the Internet. Methods of payment can be one of the following:
    - Credit card number entered directly to STB using the remote control device.
    - Previously configured credit card.
    - Previously configured payment service (e.g. PayPal or Google Checkout, etc.) account and password.
4. Auction purchases: Subscriber buys items from eBay (or uBid, etc.). Payments can be made using interaction with the auction site.
5. Interacting with Social Networking sites: e.g. Twitter, Facebook, etc.

6. Games: Subscriber can buy games, or playing time, and compete against other subscribers. The different categories of games may be:
   - Single player games (like Sudoku),
   - Multi-player games (like Chess, Checkers), or
   - Team player games, where intra team communication is needed between players.
7. An MSO may receive MPEG multiplex feeds from various national sources, such as ESPN, Disney, etc. MSOs may also receive feeds and advertisements from local affiliate sources. These feeds may contain embedded eTV applications.
8. MSOs may receive feeds from a wide variety of sources, where some sources are less reputable than others. The application ID and publisher ID may need to be validated for authenticity before the eTV application is run on the STB.
9. Applications may have varying authorization requirements. For example, some applications may need to display widgets that are "on top" of other widgets, like CallerID displays or emergency notifications.

10.3 Solution Constraints

This section lists various assumptions and constraints that place limits on the spectrum of possible solutions.

1. The solution should be lightweight:
   a. low processor overhead, low memory overhead,
   b. low maintenance and administration overhead,
   c. easy to program.
2. The solution should support low end devices.
3. The solution should assume that the legacy network may not be secure.
4. The solution should be "Reasonable" and best effort security:
   a. PKI and DES not suitable on low-end STBs,
   b. hardware encryption and key protection is not a requirement.
5. The solution should leverage CAS keys that are already present on STBs.
6. The solution should free or low-cost licenses for required technology.
7. The User Agent is assumed to be trusted as this is a proprietary platform tied to the STB firmware.
8. The Communication between the User Agent and the Head-end is assumed to be sufficiently complex to offer a small amount of protection for the initial use cases. However, as applications become more powerful, this communication will need to be secured further.

10.4 Threat Analysis

This section describes potential attack points and threats to the eTV service.

The main attack points are:

1. Before the application reaches the MSO network from the affiliate.
2. While the application/data is flowing through the MSO network to the STB.
3. While the application is running on the STB.
4. While data is flowing back from the STB through the MSO network and to outside networks.

Some of the potential threats are listed below:

1. An unauthorized person in the household may order services or items.

2. An unauthorized neighbor may order items from a home shopping channel using the subscriber's account.
3. PINs, credit card information, or other sensitive information may be sniffed while being sent across the proprietary or public networks.
4. Sensitive information like credit card numbers, PayPal account information, etc. may be accessed by unauthorized persons.
5. Information like credit card numbers may be compromised when a STB is recycled, or permanently lost if STB fails.
7. An unauthorized person may request information to be sent to a subscriber, thereby spamming the subscriber.
8. Subscriber's request for sensitive information may be disclosed, leading to loss of privacy and anonymity.
9. An application may corrupt sensitive information left behind by another application.
10. An application may consume too much resources, thereby affecting other applications.
11. Subscriber may try to adversely affect the results of a vote with multiple votes.
12. Subscriber's vote may be disclosed, leading to loss of privacy and anonymity.
13. Game player may attempt to modify parameters to modify score or rewards.
14. Denial-of-Service attacks to block games, or intra team communications.
15. Player may attempt to use DVR or Time Shift Buffer to gain game rewards or credits, for example, by replaying moves.
16. Unauthorized snooping on rival team's communications.
17. Unauthorized spoofing of intra team communications.
18. Unauthorized modification of saved game state or game credits.
19. An unauthorized player may resume a different player's saved game.
20. Unauthorized access of Social Networking sites.
21. Feeds may contain embedded EBIF applications. Some of these applications may not be authorized for distribution due to licensing issues. Filtering out all the applications in a feed may not be acceptable if some applications are authorized and others are not. This may lead to loss of revenue.
22. An application may be improperly constructed to masquerade as a licensed or authorized application by forging IDs.
23. An application may be unsafely written or untested, and therefore unauthorized for broadcast.

10.5 Security Architecture

<Note: this section is not complete, currently security architecture is under review and development.>

This section describes the high level architecture which employs a layered approach:

1) Carriage Agreements

Agreements with affiliates regarding what applications will be carried by MSOs on their network need to list remedies if an application acts maliciously.

2) Security guidelines to application vendors

Under the current architecture supported by legacy devices, very little security support is available for applications for processing or storing sensitive information like customer credit card information, and additional constraints apply as listed in Section 10.3. Application vendors are encouraged to design applications to run securely within these constraints.

3) PID Management

MSOs will have the ability filter traffic as outlined in Section 5.6.

4) Monitoring and control and by User Agent

MSO User Agents can offer the last layer of control by extending the stewardship function to the STB; for example by maintaining a “white list” of good applications, and not allowing applications that are not in the white list to launch. This approach is outlined under the Section 7.3.2.3, Client Validation.

11 EXECUTION ENVIRONMENT

This section describes the runtime environment for enhancements and their access to device resources.

11.1 File System Access

This section will be completed in a future release of this document.

11.2 Security of Viewer Input

The run time on the client (set-top device) should be inherently secure to ensure that the payload sent to it, is a predictable response. For example, how is it ensured that the response to an on screen prompt (e.g., Poll questions) is truly based on the user pressing a remote key. On OCAP/DOCSIS there is lots of security but with legacy set-top devices this is a bit of a challenge. Where/how/by whom does this get addressed for legacy set-top devices?

11.3 Application Permissions

Application Permissions will be defined in a future release of this document.

11.4 Discovery of Platform and Network Configuration

This section will be completed in a future release of this document.

12 TESTING

This section is largely based on the proposed OCAP Host and Application Interoperability Testing process, adapted to the ETV environment. It should be noted that this approach may only need to be applied to native User Agents, as OCAP-compliant agents could be considered as special cases of OCAP application testing.

It is assumed there is a three-way relationship between host testing, application testing, and interoperability testing.

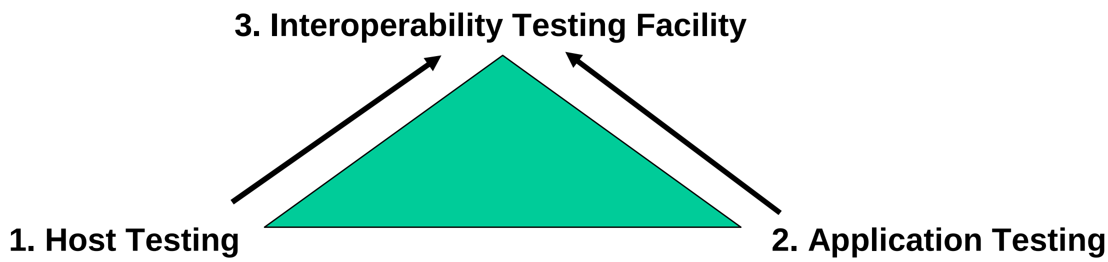

Figure 12–1 - Testing Relationships

Here a Host means a native ETV User Agent bundled with a legacy set-top box.

An RTF (*Required Test Facility*) provides the environment (limited in size), wherein both hosts and applications must successfully pass interoperability tests in order to go to the field.

Hosts and apps may optionally go to a Zoo (no size limit) for voluntary testing.

The following table summarizes the proposed steps of the overall test process.

Table 12–1 - Testing Steps

		Hosts	**Applications**
Prerequisites	Hardware	Self-tests	Self-tests
	Software	No separate User Agent certification required	EBIF Application or Authoring Tool Validation by CableLabs or 3rd Party
Required Interoperability Testing	Certified Host/Application Interoperability	Test new Host using the set of Apps and App Tests in the App RTF	Test new app using the set of Hosts and App Tests in the Host RTF
Voluntary Interoperability Testing	Voluntary Host/Application Interoperability	Test new Host using the set of Apps and App Tests in the App Zoo	Test new app using the set of Hosts and App Tests in the Host Zoo

Figure 12–2 is a flow diagram for the process.

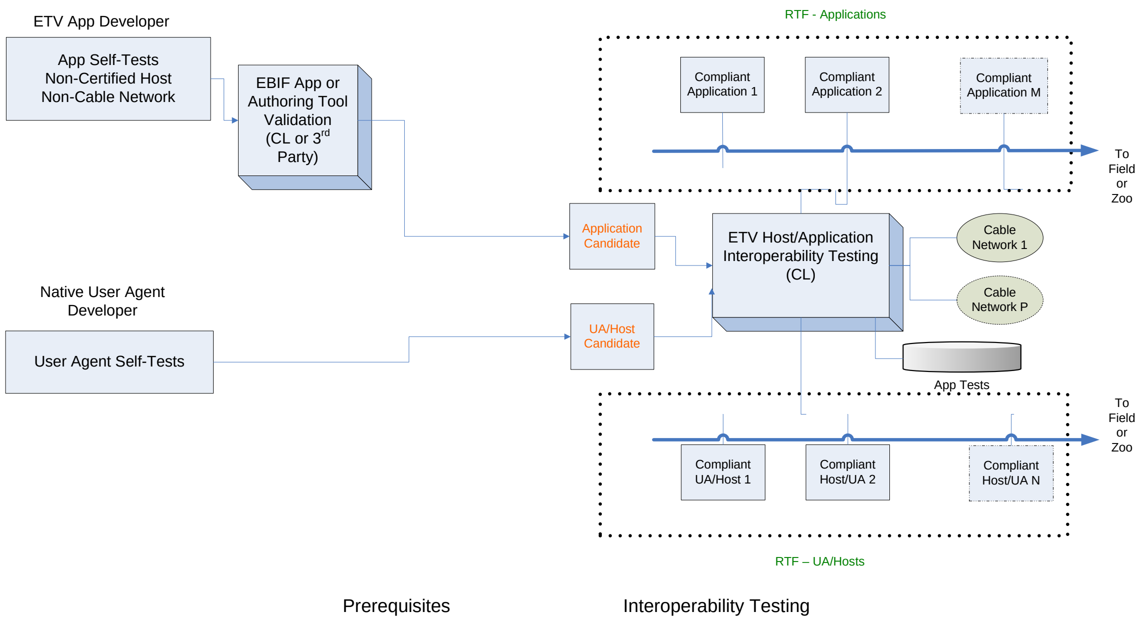

Figure 12–2 - Testing Process Flow

The following sections outline the details of this process flow.

12.1 Host and Application Self-Tests

It assumed that the developers of ETV applications and user agents will have their own set of functional tests that will be executed prior to entering any formal (i.e., certified) test process.

12.2 EBIF Application Validation

This step ensures that the target output stream containing the application commands and data is EBIF-compliant. There are several existing commercial products that provide this function.

As an alternative, the authoring tool that produces the EBIF output could itself be certified. EBIF apps generated by a certified authoring tool could have a tool-certification stamp embedded in them, guaranteeing the validity of the contents.

12.3 App RTF

The App RTF (Required Test Facility) is a limited set of agreed-upon ETV applications and their associated tests ("App Tests") that that have previously passed interoperability certification, and which fairly represent what a new ETV Host should support.

Initially the App RTF will be seeded with a few representative applications.

12.4 Host RTF

The Host RTF is a limited set of agreed-upon ETV hosts that have previously passed interoperability certification, using the Apps and App Tests from the App RTF, and fairly represent the wider population of all ETV hosts.

Initially the Host RTF will be seeded with a few representative hosts.

12.5 App Tests

The App Tests are modeled on the complete functional tests associated with each App in the App RTF, but pared down to focus on the stress, boundary and edge cases of those tests. The likely source of those complete functional tests is the app provider's self-tests.

App Tests will be largely manual at first, with automation as a long-term goal.

12.6 ETV Host/Application Interoperability Testing

This process step consists of running the App Tests on a new host or application candidate, the output of such testing being App Test Scores. These scores are used as the criteria for interoperability certification of the host or application candidate.

Applications and hosts that pass certification may also be considered for membership in the App and Host RTFs.

13 VERSIONING

This section highlights the versioning considerations related to delivering a complete end-to-end ETV experience. It focuses on four specific areas:

- Signaling
- Platform
- Return Channel Messages
- Testing of New Versions

13.1 Signaling

Versioning manifests itself in two areas of signaling:

- Protocol Version Number
- Application Version Number

13.1.1 Protocol Version Number

Per the specification, only one EISS stream may be delivered per MPEG-2 program. An EISS stream includes a major and minor protocol version that the user agent is expected to use for validation. As the specification is updated, a change in this value needs to be anticipated.

The CableLabs Operational Guidelines should provide guidance to application developers, infrastructure providers and user agents on the requirements for supporting protocol versions. Specifically:

- Date at which support for a major protocol version is required;
- Date at which support for a prior major protocol version may be deprecated.

This implies that there is some time window during which vendors are required to support multiple protocol versions being signaled simultaneously in a broadcast stream.

Some options for ways for the content provider, MSO and solution providers to approach this are to:

- Wait until all clients are updated;
- Carry 2 versions of signaling until all clients are updated. This approach will require the control of delivery using:
    - Affiliate/provider controlled equipment;
    - MSO controlled equipment.

13.1.2 Application Version Number

The EISS application information descriptor carries an application_version() field that indicates the major and minor application version.

There is a coordination task between the content provider and the MSO when the version of an application is changed. If the MSO is applying a filtering solution in the acquisition network to control what it passes to subscribers, an update to an application version could result in loss of enhancement in the absence of coordination.

A change in application version may also imply an update to external resources the application utilizes.

13.2 Platform

In most cases an application and its associated resources will be designed and delivered so it can be processed on all hardware and software platforms. The ETV specification does recognize that different hardware and/or software platforms may provide a different level of functionality (ex. DVR enabled set-top device). Special emphasis is placed on legacy set-top box models that may be limited in a number of areas versus newer set-top box models. Each resource can be tagged with a platform directory table that allows the EBIF engine to choose the most appropriate resource to load for a given application. The platform directory table includes both hardware and software version tags. A generic tag can be specified as a default.

The signaling [ETV-AM] and format [ETV-BIF] specifications include the processing details.

The impact of platform-level filtering requires close coordination from the content provider community and the MSO when new software or hardware is deployed. The following use cases will require coordination:

- New set-top device becomes available at retail
- MSO procures a new set-top device that is made available to subscribers
- A set-top device receives a software upgrade

13.3 Return Channel Messages

For applications that utilize client/server communication updates to the application, messaging will require careful coordination. The content provider must recognize that content that uses both old and new messaging constructs may exist for an extended period of time. An application may be saved on a DVR, embedded in an enhanced advertisement, embedded in a piece of VOD content, etc.

The return channel communications can be affected by version changes in:

- Change in server components
- Change in client enhancement/application versions
- Change in server applications

Content providers/developers should anticipate these changes and plan to accommodate them into their server side architecture.

13.4 Testing of New Versions

In general there will be an overlap period when a new version of an application is being deployed while the previous version must still be supported. Both versions may appear in the same transport stream simultaneously.

It has been suggested that applications make use of the "Testable" flag such that user agents would only launch the new version if they had been instructed by some external notification to check this flag and if the flag were set. Unless both conditions apply, agents would only launch the old version.

14 CONTENT SPLICING

Enhanced advertising must be supported by systems that support the schedule and play of local advertising. These systems typically store local advertising on Ad Servers and insert these ads into network feeds that are delivered via satellite or on terrestrial networks. The splicing of local ads into network feeds is accomplished by Ad Servers and splicers; these devices are coordinated by standard APIs as described in Section 14.2 below.

The following sections outline the requirements for splicing ETV content and give an overview of linear ad insertion.

These introductory sections are followed by guidelines for ETV splicing in two specific system settings:

- Certain existing set-top boxes (henceforth called *legacy* clients) have limited capabilities in terms of the dynamic introduction of new data tracks. These are clients that support only the Baseline ETV Profile. To support these clients, splicers must maintain a fixed set of baseline ETV data tracks. This will be described in Section 14.3.
- Newer clients will support a richer set of ETV content conforming to the Full and Advanced ETV Profiles, Section 14.4 will discuss the ramifications when these profiles and present guidelines for splicing multi-track ETV content.

14.1 Splicing Requirements

The following are the basic requirements for splicing ETV content.

- Support for a broad set of clients. Both existing legacy, clients and potential advanced clients should support the splicing of local advertisements. That is, the splicing guidelines presented here should support Baseline, Full, and Advanced User Agents. It must be possible to support a mixed environment of all ETV profiles.
- Quality of Transitions. The presence of ETV data tracks should not impact video and audio transitions. No ETV data should be lost in the splicing process.
- Security. The splicing system should not facilitate ad killers; nor should the ETV data tracks compromise any existing encryption models.
- Transparency. The processing of dynamically spliced streams should be identical to the processing of these streams in the absence of dynamic splicing.
- The guidelines presented here should leverage SCTE 30, 35, and 67. Ideally modifications to these standards will not be necessary but will be requested through SCTE DVS Working Group 5 if needed.

14.2 Linear Ad Insertion Overview

The SCTE DVS Working Group 5 has promoted a number of standards for coordinating the timing and placement of local advertisements in network feeds:

- [SCTE 35] is used as a standard way to mark locations in network feeds that indicate the beginning and end points of national advertisements. These marked advertisements are available for replacement by local ads. SCTE 35 was designed to support MPEG data tracks in addition to video and audio and are consistent with the standard use of data tracks specified in the ETV standard.
- [SCTE 67] serves as an information enhancement to SCTE 35.
- [SCTE 30] is used to coordinate the actions of an ad server with the arrival and processing of network feeds by a splicer. This coordination is based on the synchronization of clocks on the server(s) and splicers, which is typically accomplished using Network Time Protocol (NTP). SCTE 30 is designed to support the splicing of video, audio or data tracks and supports ETV content.

The basic flow of content and control is shown below:

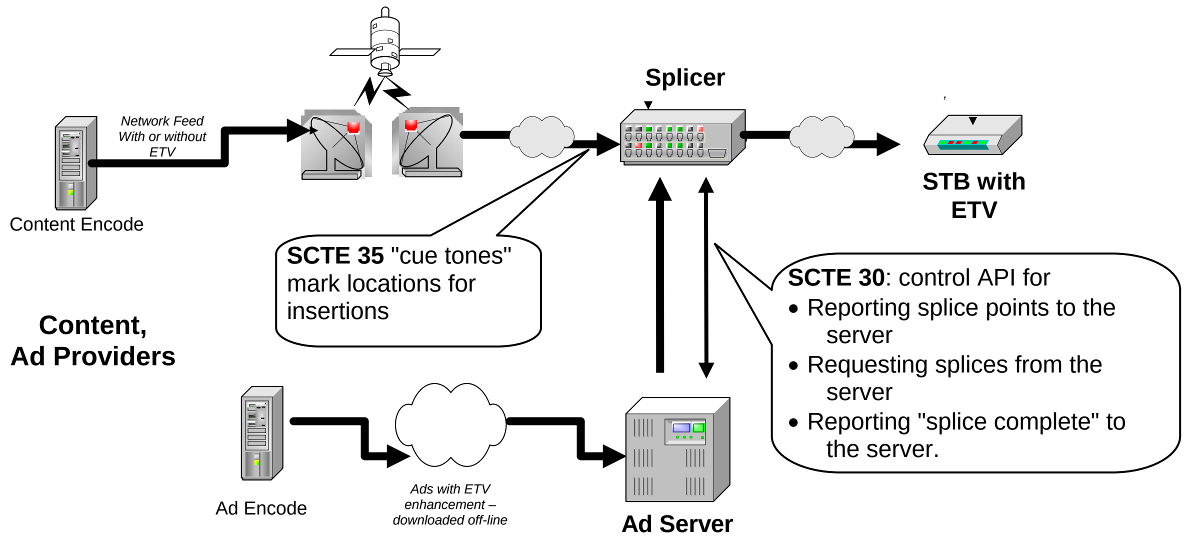

Figure 14–1 - Linear Ad Insertion

The flow of SCTE messages is show below.

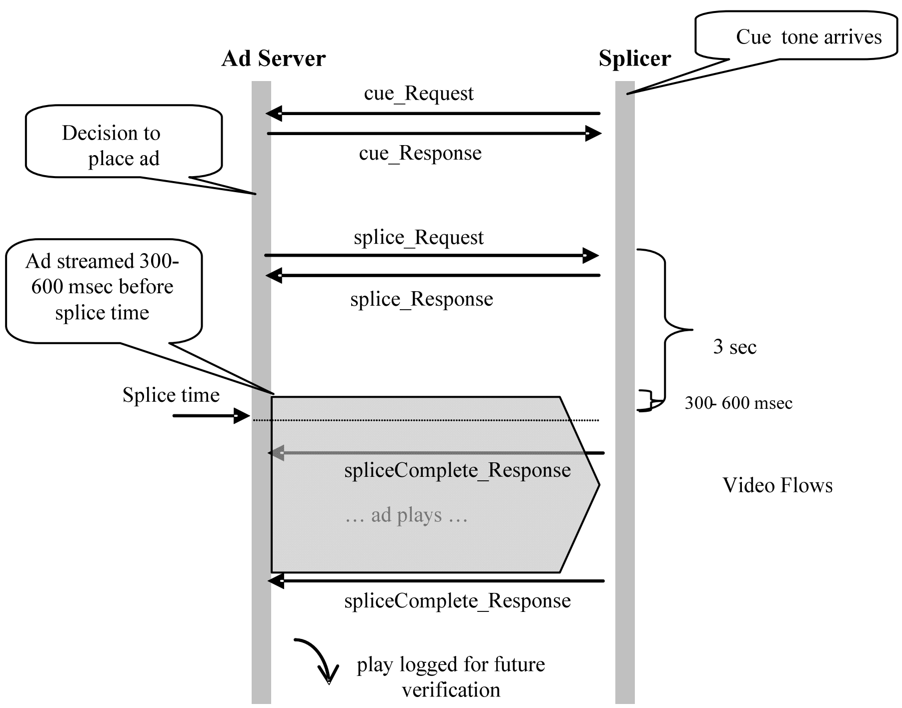

Figure 14–2 - SCTE 30 Messaging

As mentioned above, the data tracks specified in the ETV standard, the EISS and the ETV BIF tracks, can be signaled as generic data tracks. Specially,

- The Cue-Request message contains the incoming SCTE 35 splice_info_section which in turn contains the splice_time (in PTS units) for the splice point. Note that splicers are at liberty to select splice points based on this value but not necessary identical to this value. Thus an actual splice point may vary by a small amount. For this reason it is recommended that ETV data should not occur within 500 msec of a splice point.
- The Splice_Request message can identify a splice by either ServiceID (i.e., program_number) or on a component-by-component basis. This component-based style is required with splicing ETV streams. That is, when splicing ETV streams, the ServiceID must be 0xFFFF. This requires the listing of each pid and stream_type of the elementary streams.

  With the ServiceID=0xFFFF, the Splice_Request includes an embedded structure called the splice_elementary_stream() which, in turn, contains the descriptors contained in the ad's PMT. These would, therefore, contain the etv_registration_descriptors, the etv_integrated_signaling_descriptor, and the etv_bif_platform_descriptor described in the ETV specifications.

  All other fields in the Splice_Request are unrestricted and described in [SCTE 30].

Two issues are addressed in the following two sections. The first addresses legacy set-top boxes supporting the baseline ETV profile; the second section addresses multi-track ETV applications which are supported by Full and Advanced profiles.

Note that these are not mutually exclusive cases. Splicers can be configured to simultaneously support both baseline, full, and advanced clients.

14.3 PMT Continuity for Baseline Clients

A number of existing set-top boxes do not continuously monitor for PMT version changes. Under these circumstances, it is possible that an ad containing ETV data would not be processed nor would have excessive delays in processing the enhancement relative to the duration of the ad. The solution to this problem is to maintain, at all times, a constant set of pids with fixed descriptors in the PMT to be processed by legacy clients. These shall be called *baseline pids* representing *baseline streams*.

As specified in the ETV specification, baseline ETV content will have only one EBIF track. Thus there will be exactly two baseline ETV streams, the ETV Integrated Signaling Stream (EISS) and the single EBIF baseline stream.

When the baseline PMT entries do not exist in the network feed, the splicer must include them in its output. This is a configuration operation on the splicer. That is, it must be possible to configure a splicer with outputs that are designated "baseline ETV outputs". When configured in this way, the splicer will stream PMTs on these outputs with baseline pids and descriptors specified below in Sections 14.3.1 and 14.3.2 below.

In general other full or advanced profile elementary streams may change dynamically through the standard PMT versioning mechanism. Note that in some systems, the dynamics of non-baseline streams may also be restricted. Such restrictions are based on system requirements and are outside the scope of the ETV specification.

For baseline ETV streams, the following structures shall be used. Note that these are constant structures that do not change from site-to-site or over time. This allows splicers to be initialized easily to support ETV content in such a way that PMTs do not change and legacy set-top boxes do not have to respond to changes in PMTs.

14.3.1 ETV Integrated Signaling Stream

The ETV integrated signaling stream component of the Splicer Output Channel is described by two PMT descriptors for component stream type 0xC0, as follows:

Table 14–1 - ETV Integrated Signaling Stream ETV registration descriptor

Syntax	Bits	Data
`etv_registration_descriptor() {`		
`descriptor_tag`	8	0x05
`descriptor_length`	8	0x04
`etv_format_identifier`	32	0x45545631
`}`		

Table 14–2 - ETV Integrated Signaling Stream ETV integrated signaling descriptor

Syntax	Bits	Data
`etv_integrated_signaling_descriptor() {`		
`descriptor_tag`	8	0xA2
`descriptor_length`	8	0x00
`}`		

14.3.2 ETV Resource Stream

The ETV resource stream component of the Splicer Output Channel is described by two PMT descriptors for component stream type 0xC0, as follows:

Table 14–3 - ETV Resource Stream ETV registration descriptor

Syntax	Bits	Data
`etv_registration_descriptor() {`		
`descriptor_tag`	8	0x05
`descriptor_length`	8	0x04
`etv_format_identifier`	32	0x45545631
`}`		

Table 14–4 - ETV Resource Stream ETV-BIF platform descriptor

Syntax	Bits	Data
`etv_bif_platform_descriptor() {`		
`descriptor_tag`	8	0xA1
`descriptor_length`	8	0x00
`}`		

14.4 Splicing Elementary Streams

This section describes the actions to be performed by a splicer with splicing ETV streams.

The general ETV standard allows multiple ETV-BIF data tracks. Therefore, in general, there may be N network ETV-BIF pids and M ETV-BIF pids in a corresponding advertisement as shown below.

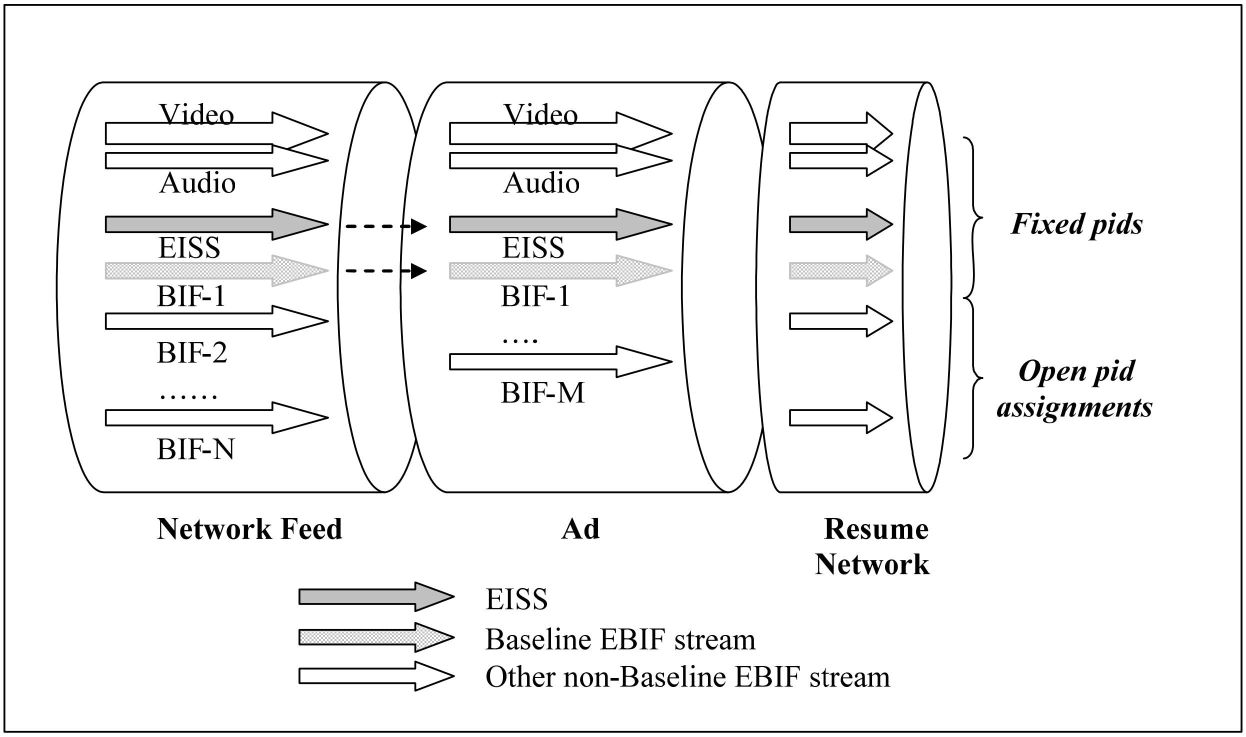

Figure 14–3 - Multiple ETV Streams

When splicing ETV streams, the following actions take place:

1. A splice_Request is issued from an Ad Server to a Splicer. The splice_Request may include any ETV elementary streams existing in the ad content.
2. The splicer will examine the pids and descriptors in the splice_elementary_stream structure and determine which stream is the EISS stream, whether a baseline EBIF stream is present, and which other EBIF streams are present.
3. The splicer will include any EBIF streams in the output that are signaled in the splice_Request message. If the splicer is configured to support baseline ETV streams, the splicer shall match the EISS stream to the EISS pid and the baseline EBIF stream to the baseline EBIF pid. All other non-baseline EBIF streams may be assigned to any pid values.
4. In general a PMT version change may be necessary. The client is expected to process the new PMT and the user agent is expected to manage any ETV transitions. If there are only baseline streams and the splicer has been configured to support baseline ETV streams, then no PMT version change shall be signaled.

Appendix I National broadcaster ETV feed scenarios

I.1 Affiliate Injection Into ETV Feed

Affiliate Local Interactive Commercials can be created using a similar methodology to that used for ETV Affiliate Application Control. The Network's interactive application is created to support a variety of local ad templates all of which the Affiliate is free to sell to sponsors. These templates contain an Ad image and an interactive component. Via the Affiliate web page, the Affiliate fills in the appropriate fields in a web page that a local advertiser has purchased depending on the template that was selected by the local advertiser. The information for this template is inserted into the Affiliate_LocalAds table that is sent via the Network streamer to the MSOs. The set-top boxes receive the Affiliate_LocalAds table and during local ads process the data on the appropriate line of the table based on the local Affiliate's call letters. A set number of "standard images" would be available with a limited number of custom images that can be uploaded from the Affiliate.

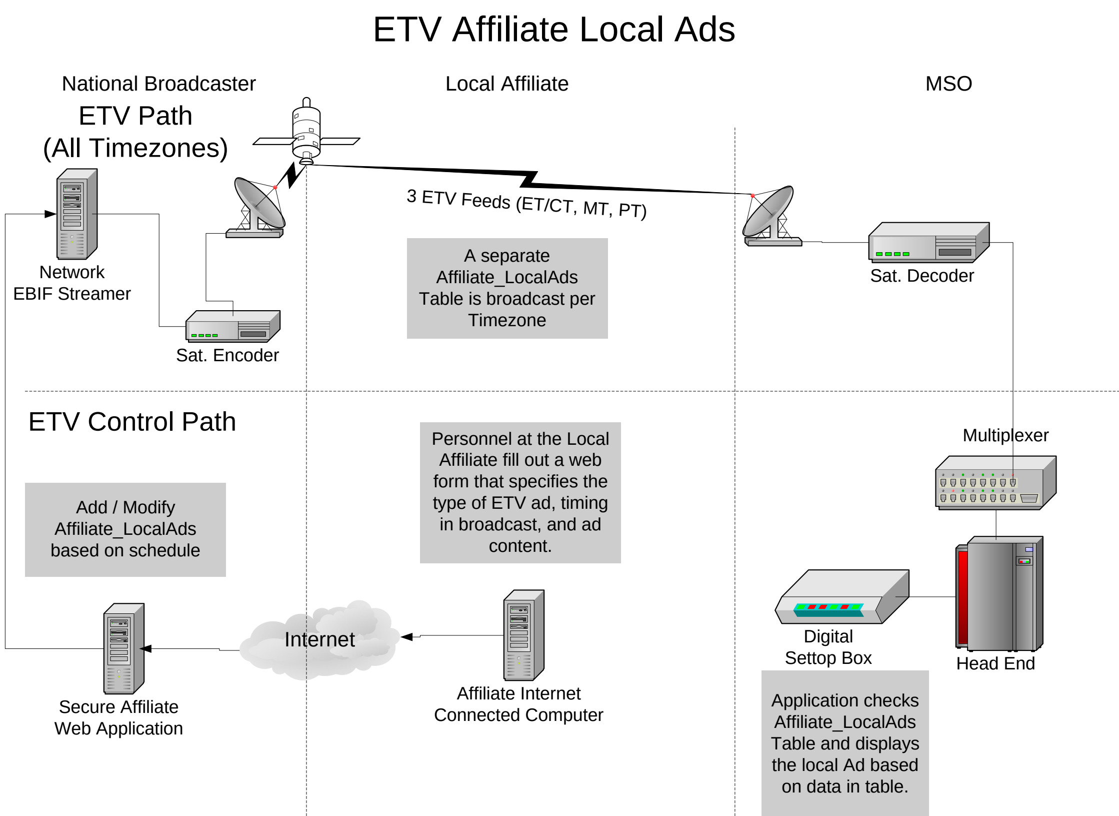

Figure I–1 - Affiliate Local Ads

I.2 Affiliate Override of ETV feed

There are times during a local broadcast when local news preempts Network delivered content or a local station will choose to broadcast local content instead of Network content. Such events create a discontinuity between the video and the ETV enhancement at the MSO level when the Enhancement is being delivered via Satellite. The solution to this issue is to request that the National Broadcasters engineer their EBIF Applications to receive an Inactive_Affiliates table which lists the affiliates who have disabled the EBIF application. This Inactive_Affiliates table would be constructed by the National Broadcaster and inserted into the EBIF satellite feed to be delivered to the Digital Set-top box. The Affiliates will be provided access to a secure web site that allows the Affiliates to disable the ETV enhancement when appropriate. This solution was selected because it allows the Affiliates to disable the enhanced feeds for all delivery mechanisms (Cable, Satellite, IPTV, mobile phone) via a single interface.

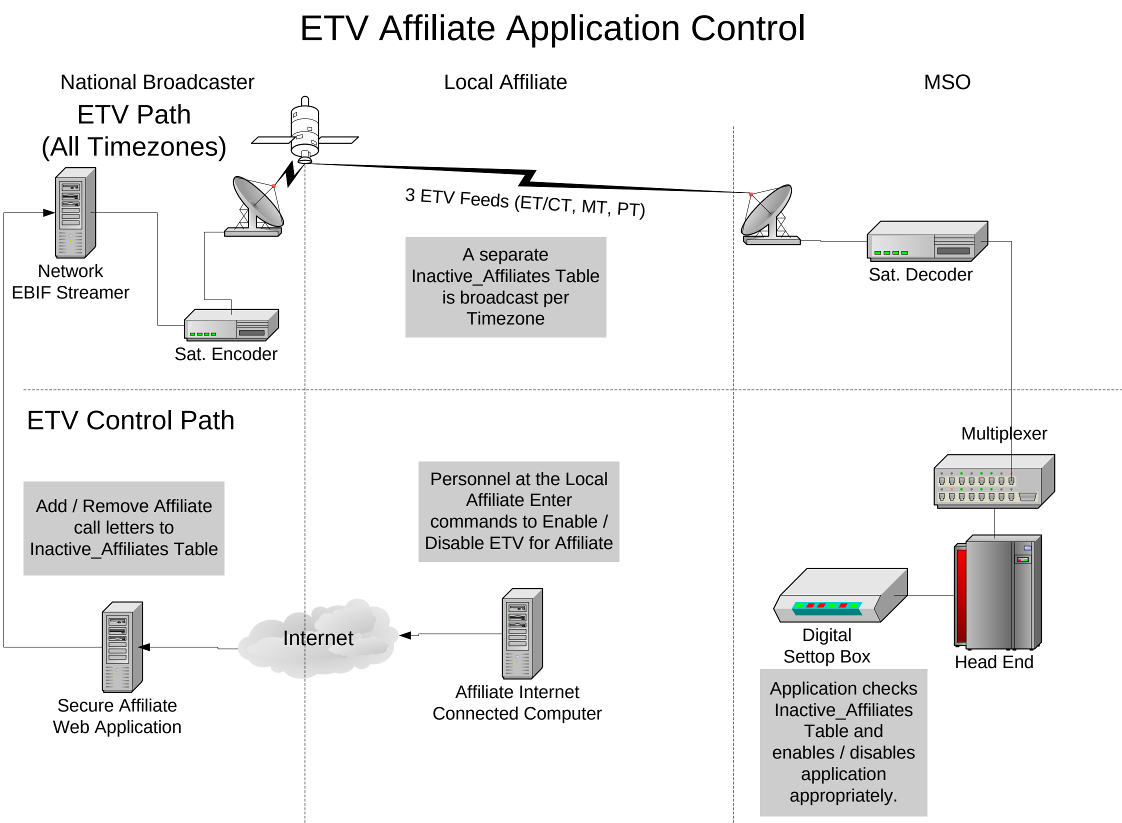

Figure I–2 - Affiliate Application Control

I.3 Mountain Time-zone issues

The solution described in Section 5.2.3.2 works for all broadcasters in all time-zones except Mountain Time. In Mountain Time, some stations add an extra minute of commercials at random times during the broadcast.

Two approaches have been identified and are described in the following subsections.

I.3.1 Standard Mountain Minute time

IF Mountain affiliates were to agree to take their minute at the same time, then the national ETV feed could be extended with a special mountain feed to accommodate the minute, and maintain synchronization.

I.3.2 Mountain affiliates Tape ETV feed

Affiliates might take the national ETV feed and inject it into their ATSC feed. Assuming MSOs support copying ETV from ATSC to NTSC programs, this mechanism maintains synchronization and allows affiliates to take their minute when they choose.

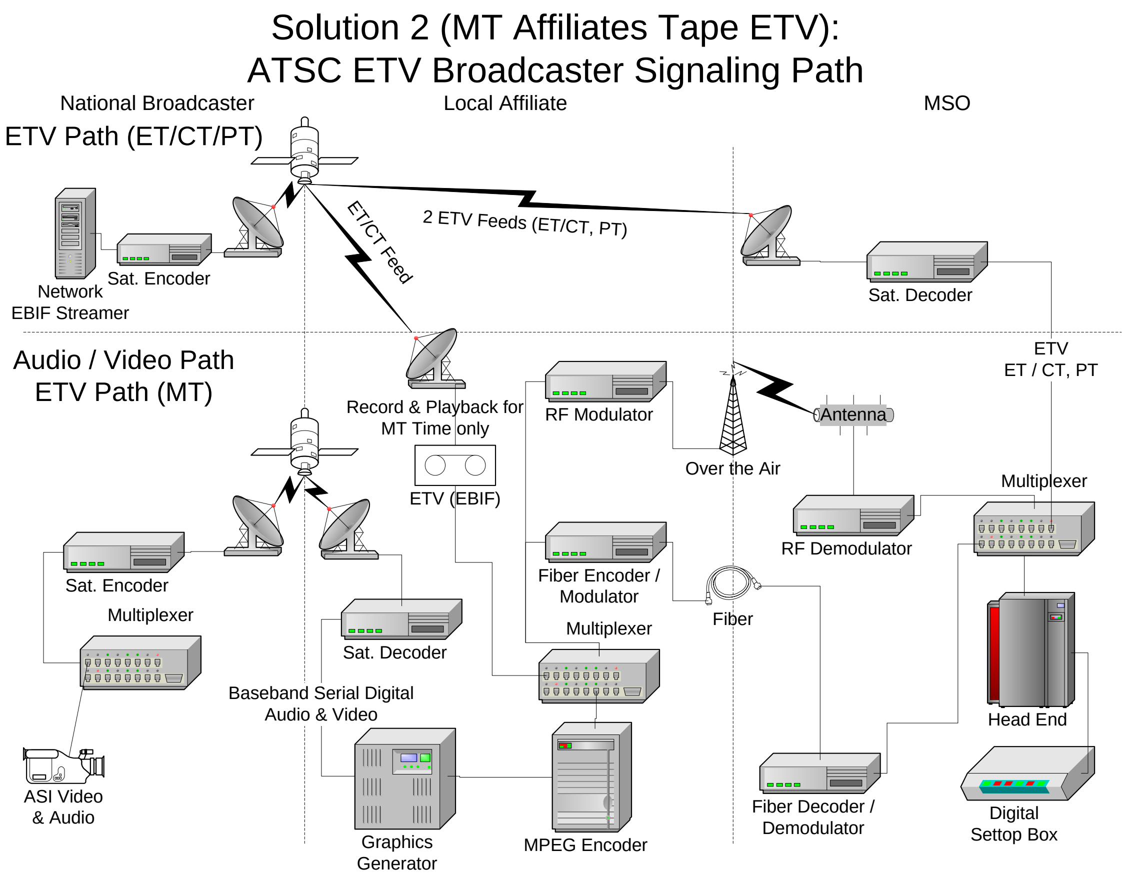

Figure I–3 - ATSC ETV Broadcaster Signaling Path. MT Affiliates Tape ETV feed

Appendix II Revision History (Informative)

The following is a summary of the main revisions incorporated into V02 version of this document:

- Replaced existing Section 5.6, MSO Management, with new section entitled Application Manager.
- Added new Section 10, Security.
- CableLabs and SCTE references updated.