Imagine COMMUNICATIONS

Broadcast Media File Interchange: Rethinking The Need For A Standard File Format

White Paper

Introduction

A transition has begun in broadcast media facilities away from traditional dedicated broadcast servers, designed specifically for storing on-air content, to more open systems that are designed to work with general purpose storage systems. This has been enabled by the development of high-performance network attached storage (NAS) systems, which have been replacing storage area networks (SAN) in the IT environment. Further fueled by the rise of file-based workflows, the demand is for more general-purpose media file support; however, the requirement to store and deliver some amount of content in an on-air, real-time manner has not dissipated.

Generally speaking, broadcast servers are designed around dedicated on-air bandwidth, corresponding to fixed numbers of ingest and playout channels and using remaining bandwidth for file transfer capability. The custom file systems designed for a specific set of often proprietary file and codec formats to ensure that on-air operation is not compromised have led to a "binary" class of quality of service (QoS): on-air and everything else. This type of design led to and enforced the concept of standardizing on specific file formats and codec operating points (e.g., a system designed with 2Gbps of bandwidth can support 40 channels of 50Mbps media) within a broadcast facility.

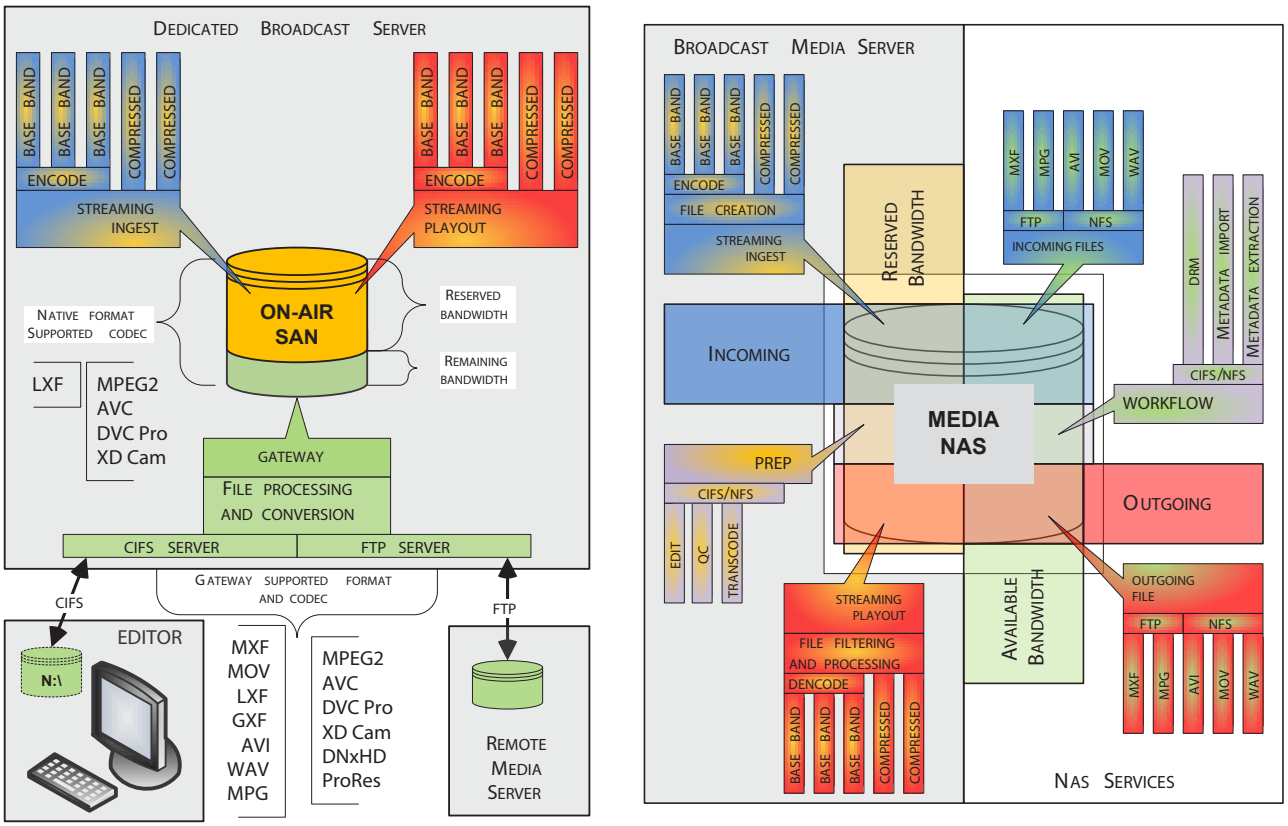
The paradigm shift has started. Basic operations such as ingest, playback, import and export are being overshadowed by a need to support advanced, distributed workflows such as collaborative editing and cloud-based media distribution. NAS and the supporting IP infrastructure systems have become more capable, codecs better and generally bandwidth more plentiful. The evolving media server needs to be more flexible in media handling, yet retain the underlying capability of correctly playing back whatever file it is given in a deterministic manner.

Server Feature	Broadcast	Media
Storage	Closed SAN	Open NAS
Design	Managed SAN for bandwidth	High performance NAS with carve out for bandwidth
Playable Media Support	Limited to known files and codecs	Limited to known files and codecs
Primary IO	Baseband streams	Files
Non-Playable Media Support	None	Any
Non-Media File Support	None	Yes (but why?)
Primary FileAccess	FTP	CIFS, NFS, FTP

Broadcast Server Evolution

The Dedicated Broadcast Server is designed around a single core goal of converting real-time streaming inputs to files, and files to streaming outputs. File to file access and operations are secondary operations designed not to interfere with this primary goal.

This type of server is usually designed around a closed, tightly managed SAN. The server provides the blockbased formatting, partitioning and allocation of storage and creates the file system for logical access to media. It also creates the access mechanism required for the various supported applications. File interchange and connectivity is hosted off the SAN by various application servers (i.e., file transfer protocol or FTP) and gateways. Media entering via a gateway typically needs to be transformed into a format natively supported by the server. This may involve reformatting the data, transcoding the essence, or collapsing a multi-element file structure.



In this type of server, the access mechanisms enforce the type of data (media) that can be stored, and prevent any kind of access that would be detrimental to performance. This requires that the gateway be capable of recognizing and parsing incoming media files, extracting metadata, and preventing damaged, incorrectly formatted or unsupported files from entering the system. Similarly, export is limited to native file types and any on-the-fly conversion supported by the gateway.

The Broadcast Media Server is the result of evolution to a more standard and open system. Facilitating this transition has been the adoption of NAS.

By transitioning from a SAN to a NAS, no longer is file system creation and management a core part of the server. Instead, the server instantiates the necessary services, including bandwidth reservation, on the open NAS system. Bandwidth for real-time broadcast operations is carved out by server components using NAS resources, while media access is distributed and shared by networking protocols such as Network File System (NFS) and Common Internet File System (CIFS), as well as FTP.

Media access operations and services range from the demand for reserved bandwidth for live ingest and broadcast playback, quick and acceptable responsiveness for editing, file transfer and buffered stream delivery, to slow and reliable where delays and slowdowns can be tolerated.

A consequence of this open design is that there is no longer a gateway managing file access to the system. In this type of system, the server includes a “media scanner” process to parse incoming files, extract metadata, populate asset databases and ultimately allow files to be played out by real-time components. These processes work in a similar manner to the “media scanners” built into most modern operating systems.

Media File Format

The Media File Format is the logical structure of the audio, video and metadata within the media files stored by the server file system. The media file format is a container that determines the following characteristics:

- File Compatibility – Devices that can interchange media.
- Supported Media Essence Types – Available codecs for audio and video media.
- Metadata Storage – Structural and descriptive metadata that accompanies the media.

Among the most popular broadcast media formats is the Media eXchange Format (MXF). MXF handles most broadcast media essence types, including vertical blanking and ancillary data (VBI / ANC). Now coming up on its fifteenth anniversary (and tenth year after publication) over 25 SMPTE standards, recommended practices and engineering guidelines, along with 6 Advanced Media Workflow Association "application specifications" for MXF, now exist (see annex 1).

Along with MXF, common broadcast media file formats include:

- QuickTime (.MOV) – a media format devised by Apple for both consumer and professional applications and adapted for broadcast use by Omneon video servers.
- GXF (General eXchange Format) – a vendor-specific format standardized by SMPTE that covers media types supported by Grass Valley servers. Amended by GXF-2, an SMPTE Registered Disclosure Document (RDD).
- LXF (Leitch eXchange Format) – a vendor-specific format used by Imagine Communications servers.
- MPEG (MPG) – the generalized container for MPEG compressed media used by former Pinnacle Mediastream servers as well as content and commercial distribution systems such as DG Systems and Pathfire.
- AVI (Audio Video Interleave) – a media container devised by Microsoft for both consumer and professional applications; adapted for professional use by some editing solution providers.

Another complication to the file format is the concept of reference files. Reference files don't actually contain media essence. They simply contain a list of pointers to the needed essence elements. Reference files are nearly impossible to integrate into simple FTP based workflows.

The final component to the file format is metadata. There are essentially four types of metadata that need to be considered: structural, descriptive, vertical interval and external referenced.

Structural metadata is derived from the actual essence (e.g., compression parameters for video, audio sample rate and bit resolution). Although seemingly important, structural metadata is usually re-generated as part of the essence parsing that takes place in the ingest process.

Descriptive metadata simply describes the media (e.g. timecode, title, description). Generally, the most important piece of metadata is the starting timecode value, because playback automation often relies on this information. In practice, less descriptive metadata is better - it really belongs in the Digital Asset Management (DAM) system.

Ancillary data (horizontal HANC or vertical VANC) data is part of the video essence that avoids the video compression process. Because it is not carried as part of the compressed essence, it must be carried elsewhere as defined by the media file format. Some file types such as MPEG, AVI and QuickTime can carry caption and subtitle data independently of the vertical interval (other VI information such as timecode and AFD is often lost). Externally referenced data can take the form of Extensible Markup Language (XML) sidecar files and are usually associated with reference file based formats.

Well-designed dedicated broadcast servers often support a wide variety of media file formats to promote media interchange and flexibility. Some servers can even import material in one format and export it into another.

Broadcast media servers, on the other hand, store files in their native format. It is up to the media scanner process to determine if a file can be decoded and played out by the base-band IO system.

In the case of dedicated servers, an external transcoder is needed to convert unsupported media to a supported file format; however, this complication adds cost, time, and generational picture degradation to the workflow. An NFS based server can incorporate this workflow step into a folder structure and files recognized by the media scanner as un-playable can be "moved" to a transcode process holding area.

Media Essence Type

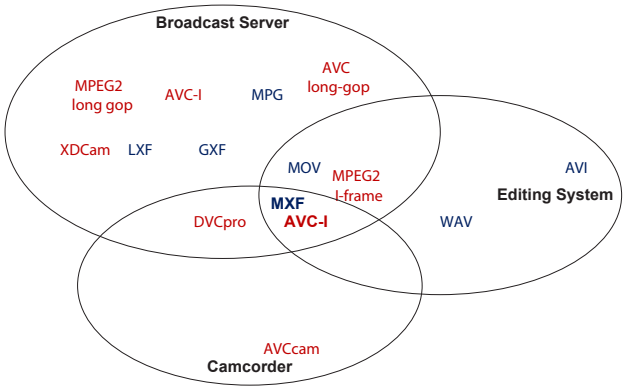
The Media Essence Type is the manner in which video, audio and vertical interval information is stored (and compressed) within the media file.

- Video – Compression formats such as MPEG2, DV, AVC and their variants
- Audio – Compression formats such as MPEG L2, ACC, Dolby-E, AC3 and uncompressed formats such as PCM and AD-PCM
- Vertical information such as analog Vertical Blanking Interval (VBI), and digital Vertical Ancillary Data (VANC) – mostly used to store timecode reference as well as captions and subtitles.

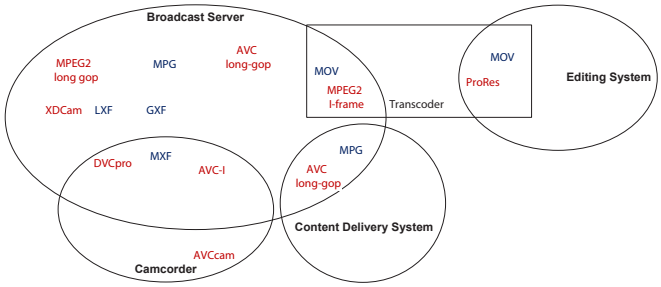
Both dedicated and media broadcast servers obviously need to have an appropriate set of codecs to decode the essence and playout media correctly. If not, a transcoding operation as previously described is required.

Design Implications

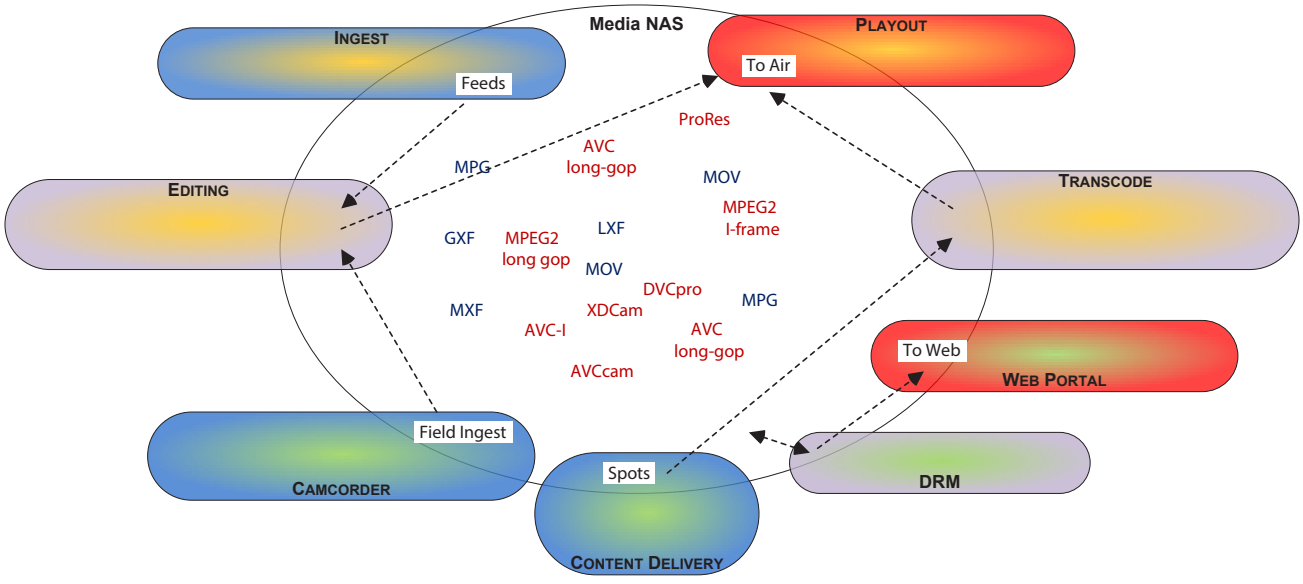
In a more traditional facility using one or more dedicated broadcast servers, it made sense to standardize on file formats and essence types that comprised the least common denominator of media systems used within the facility's workflow. While file interchange was only a subsidiary portion of broadcast operations, a simple Venn diagram was a good starting point to selecting a common format.



As it turns out, this typically works better in theory than in practice. Little inconsistencies, especially in metadata, can force the addition of transcoding to the workflow.



A dedicated broadcast server typically needs to fail a process (such as FTP) to invoke a transcode operation, which can be quite painful and time-consuming if the file is coming from a remote source. An NFS-based server can simply receive the file and flag it for additional processing. In a modern environment, this process could take place in a virtual machine or be performed by a cloud-based resource.



Conclusion

As engineers, we are impressed by the seaming ease with which consumer devices ranging from PC to tablets to smart phones can play back ANY piece of media from anywhere on the internet, at any time, with audio that mostly works and is in the right aspect ratio (aside from videos taken in portrait mode). In this world, file formats don't seem to matter; finding a device that can't upload or play video from YouTube is a challenge.

As broadcasters, we face a different reality, a world of legacy driven standards, complex types of metadata and an environment where the occasional "buffering" or "content not available" message is absolutely unacceptable. While consumers expect wider and easier access to content, our processes become more complex, cumbersome and interoperability more difficult.

Instead of fixating on specifying the ultimate, ideal file for use in a facility, it is time to analyze points of compatibility, and concentrate on removing the reasons for incompatibility-related problems from our workflows. This means requiring less metadata compatibility, picking less finicky products, not trying to adapt modern products to legacy workflows, and focusing more on the commonality between media file formats.

Imagine Communications is a global leader in video servers to media, broadcast, service provider, government, and enterprise markets. The company's scalable, interoperable video server platforms employ open standards to greatly accelerate time-to-air and dramatically reduce the costs associated with content acquisition, production, distribution and media management — today and for the long term.

Originally published in the 2014 NAB Broadcast Engineering Conference proceedings.

+1.866.4.Imagine
© 2014 Imagine Communications
Proprietary and Confidential