UMID Applications in Practice

Yoshiaki Shibata (metaFrontier.jp)
Jim Wilkinson (Wellspring Digital)

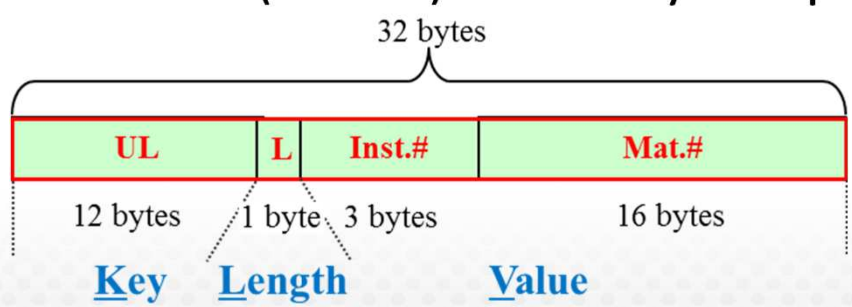
Outline

- What is the UMID ?
- UMID Application as an AV Material Identifier
  - UMID Application Principles
  - UMID Managed Domain
- UMID Applications in MXF
- UMID Resolution Protocols
- Conclusions

What is the UMID ?

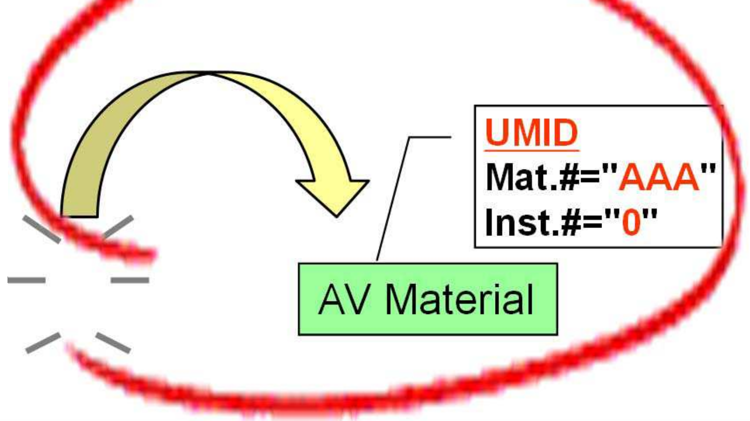
- Unique Material IDentifier
  - Specified in SMPTE ST 330:2004 & RP 205:2009
    - Universal Label(UL): identifying as UMID (Key)
    - Length(L): Length in byte that follows, Fixed to 13_h (Length)
    - Instance Number (Inst.#): Identifying "instance"
    - Material Number (Mat.#): Globally unique value

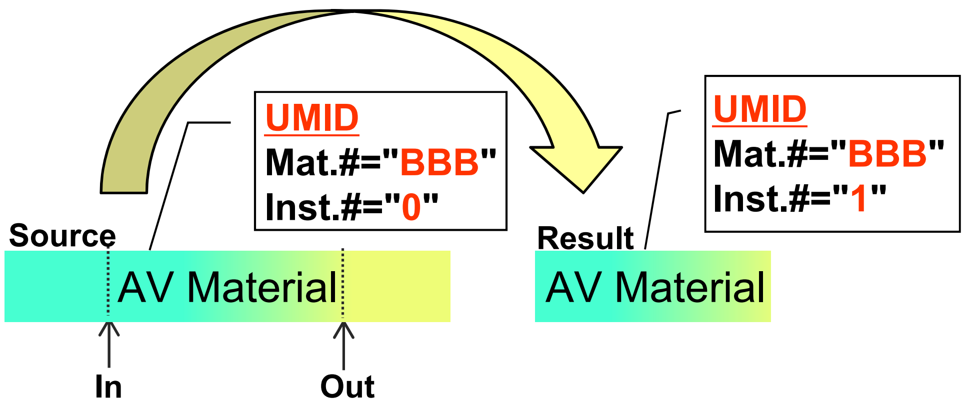
(Value)



Two Distinct Uses of the UMID

- As a unique identifier
  - Newly created value for Mat.#
  - Fixed to zero for Inst.#
- As a linking tool
  - Shared value for Mat.#
  - Non-zero for Inst.#

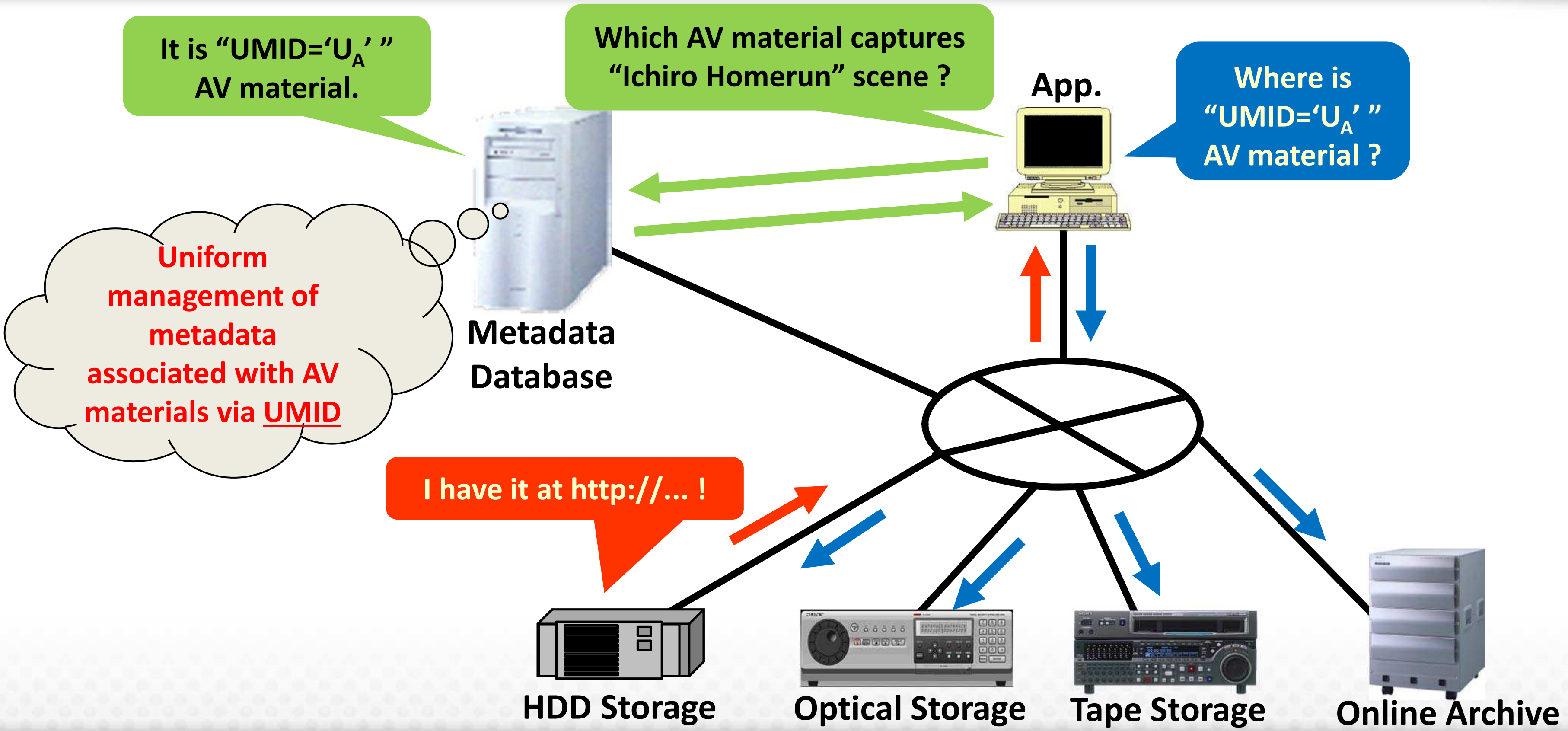




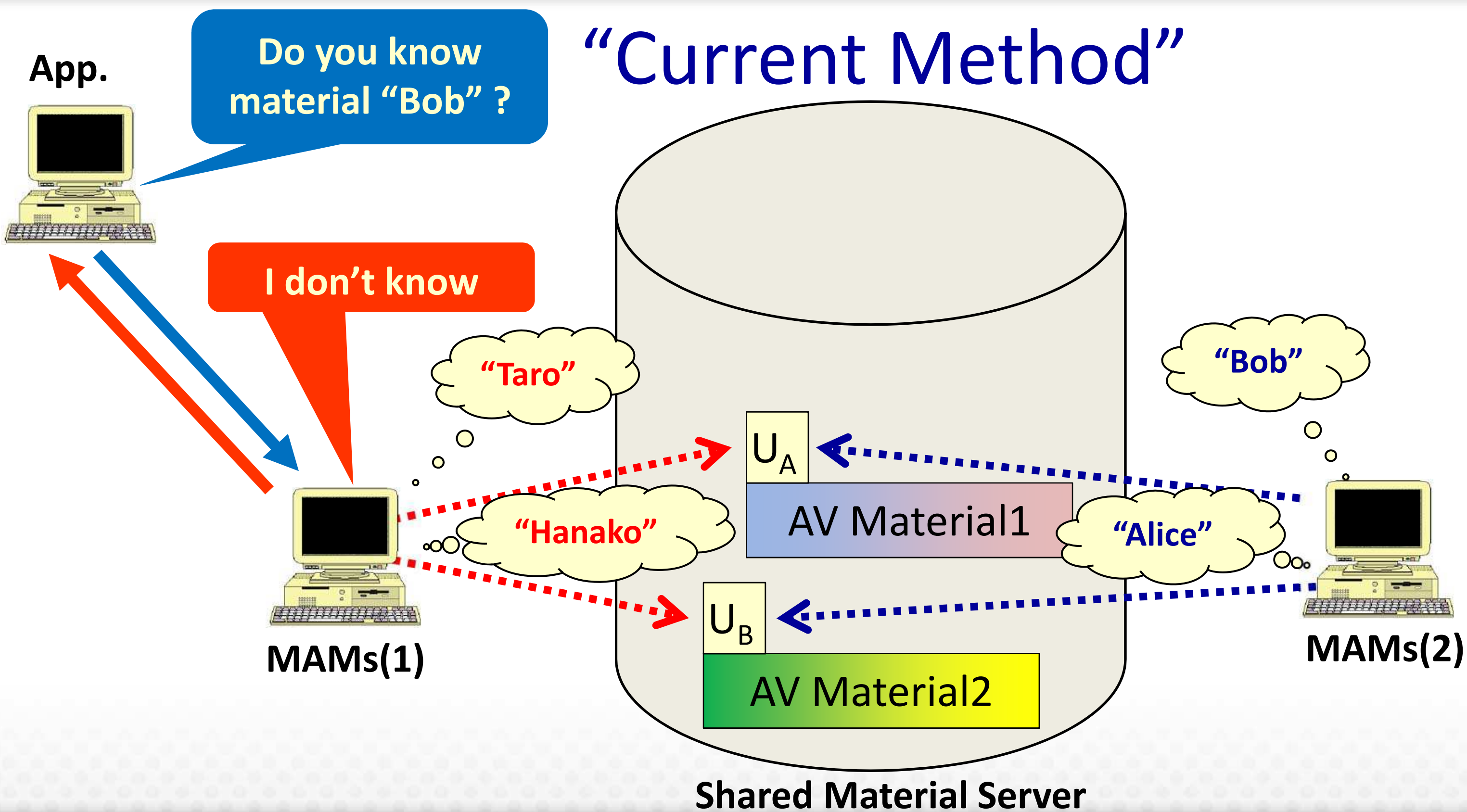
These uses are mutually exclusive !

UMID Application as an AV Material Identifier

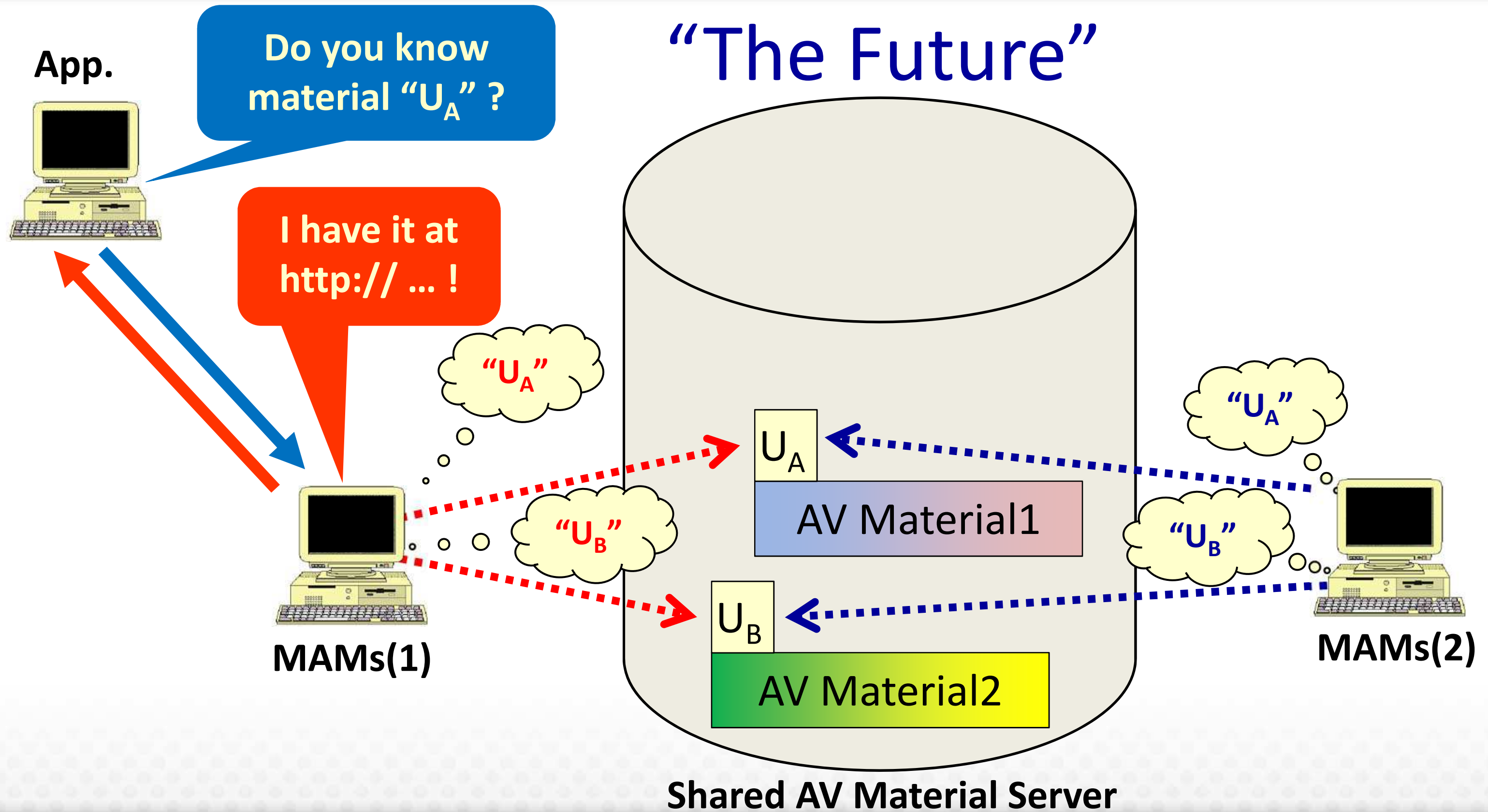
UMID based AV Material Search



Uniform AV Material Management



Uniform AV Material Management



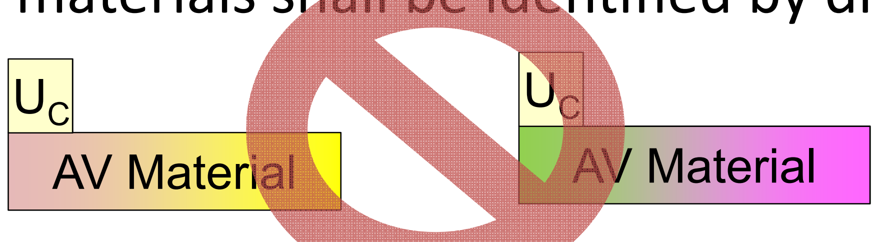
How to Achieve Them ?

1. Define industry common **UMID resolution protocol**
   - To resolves a given UMID of AV material to its URL

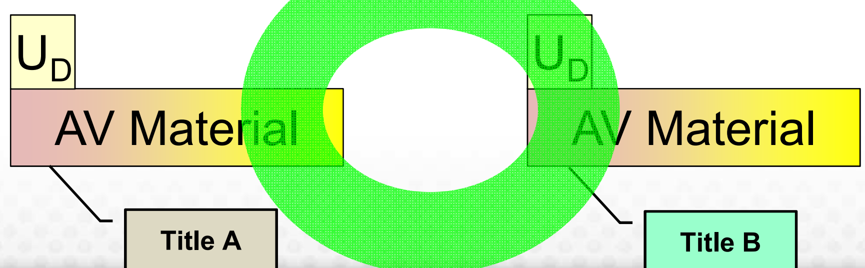
2. Recommend “UMID based AV material management”
   - After clarified **UMID Application Principles**

UMID Applications Principles

- **Principle 1: UMID Integrity**
  - Different AV materials shall be identified by different UMIDs

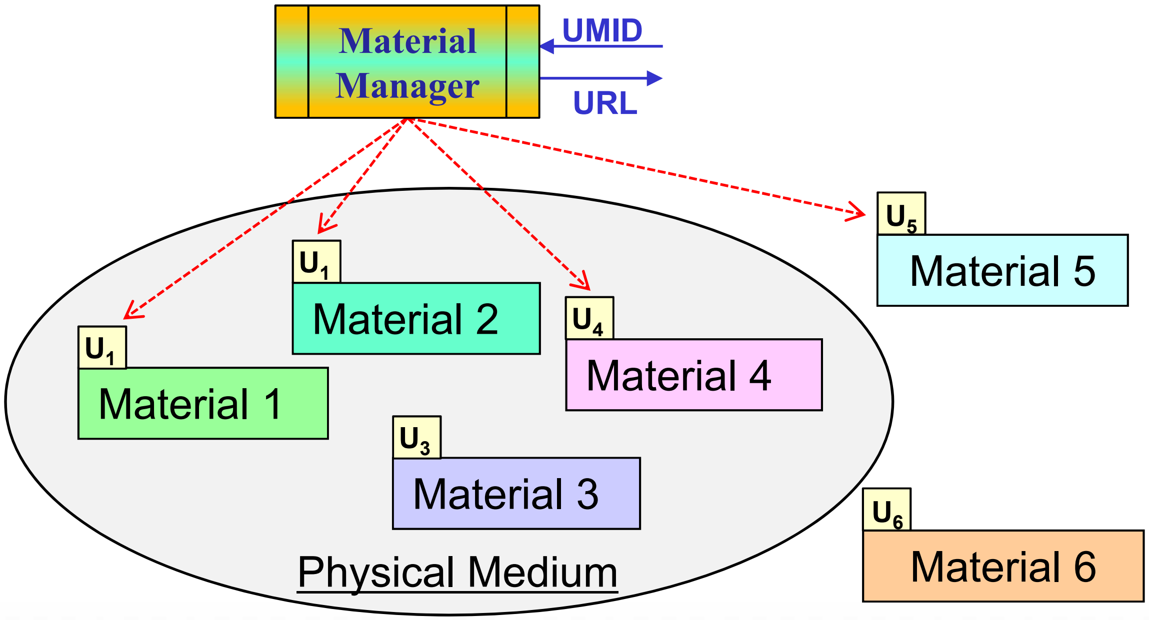


- **Principle 2: UMID Identification**
  - What the UMID identifies is the essence representation at its playout



UMID Managed Domain

- Embodiment of "UMID Application Principles"

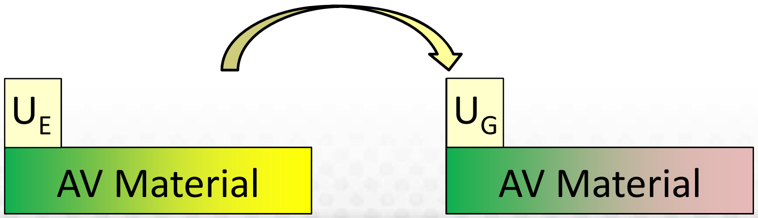


Material 1, 2, 4 and 5 constitute "UMID Managed Domain"

How Material Manager Behaves

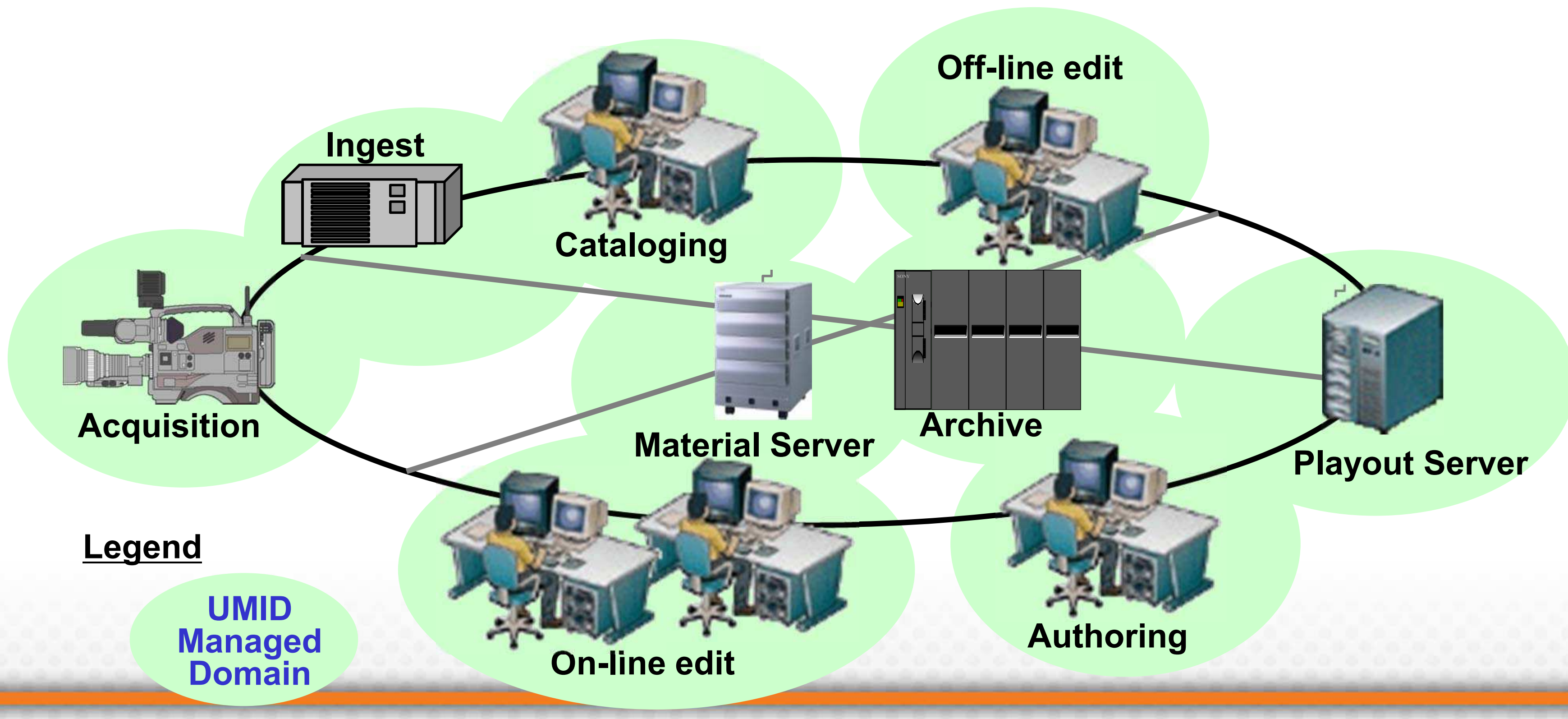
- For AV material import
  - Attach a newly created UMID
    - Except for materials from other **UMID Managed Domains**

- For AV material modification
  - Replace existing UMID with a new UMID value



UMID as Common ID in System

- "UMID Managed Domains" merged to cover a system



UMID as the Common ID in Systems

- "UMID Managed Domains" merged to cover a system

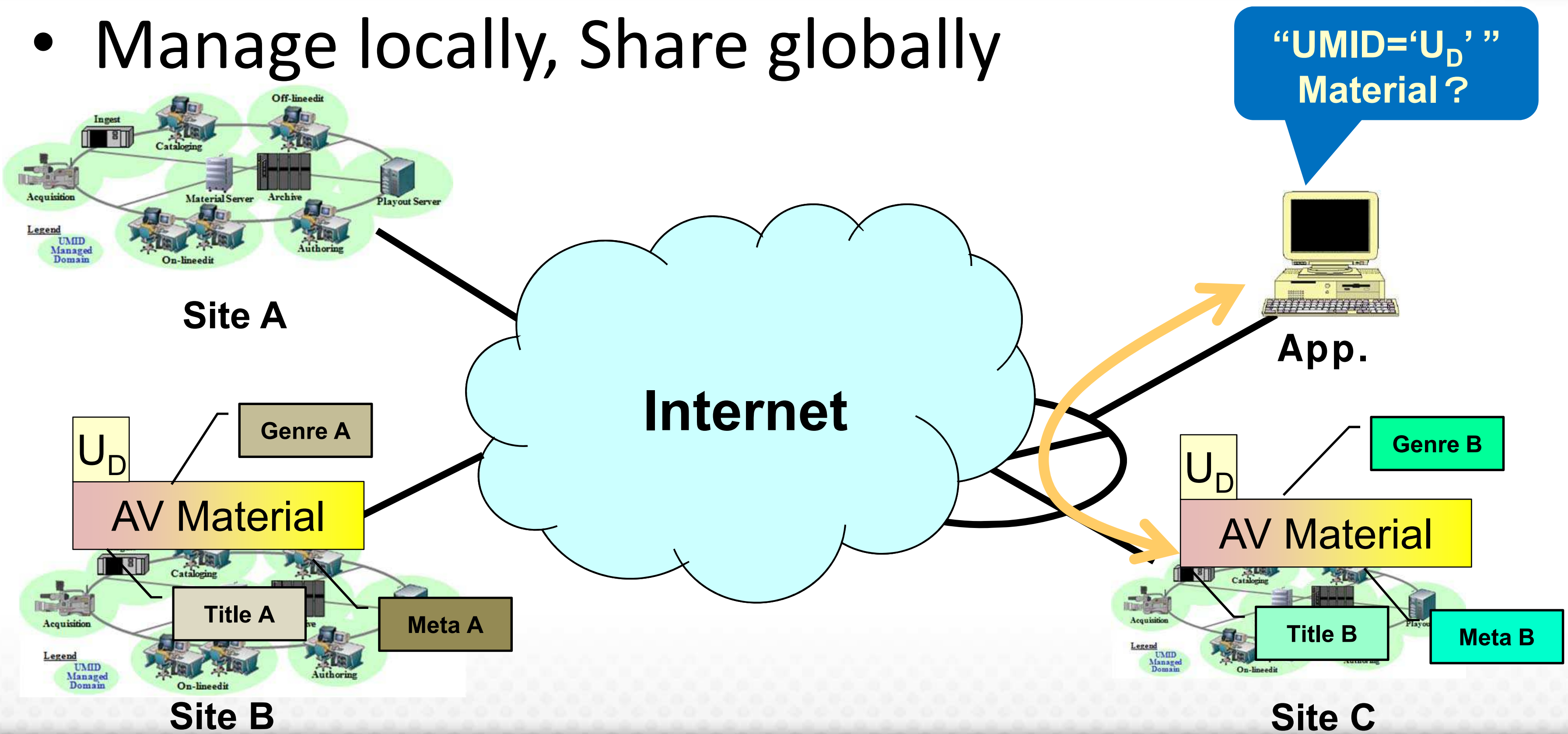
Media IT Infrastructure for

Media SOA / FIMS

(loose coupling between business and media layers)

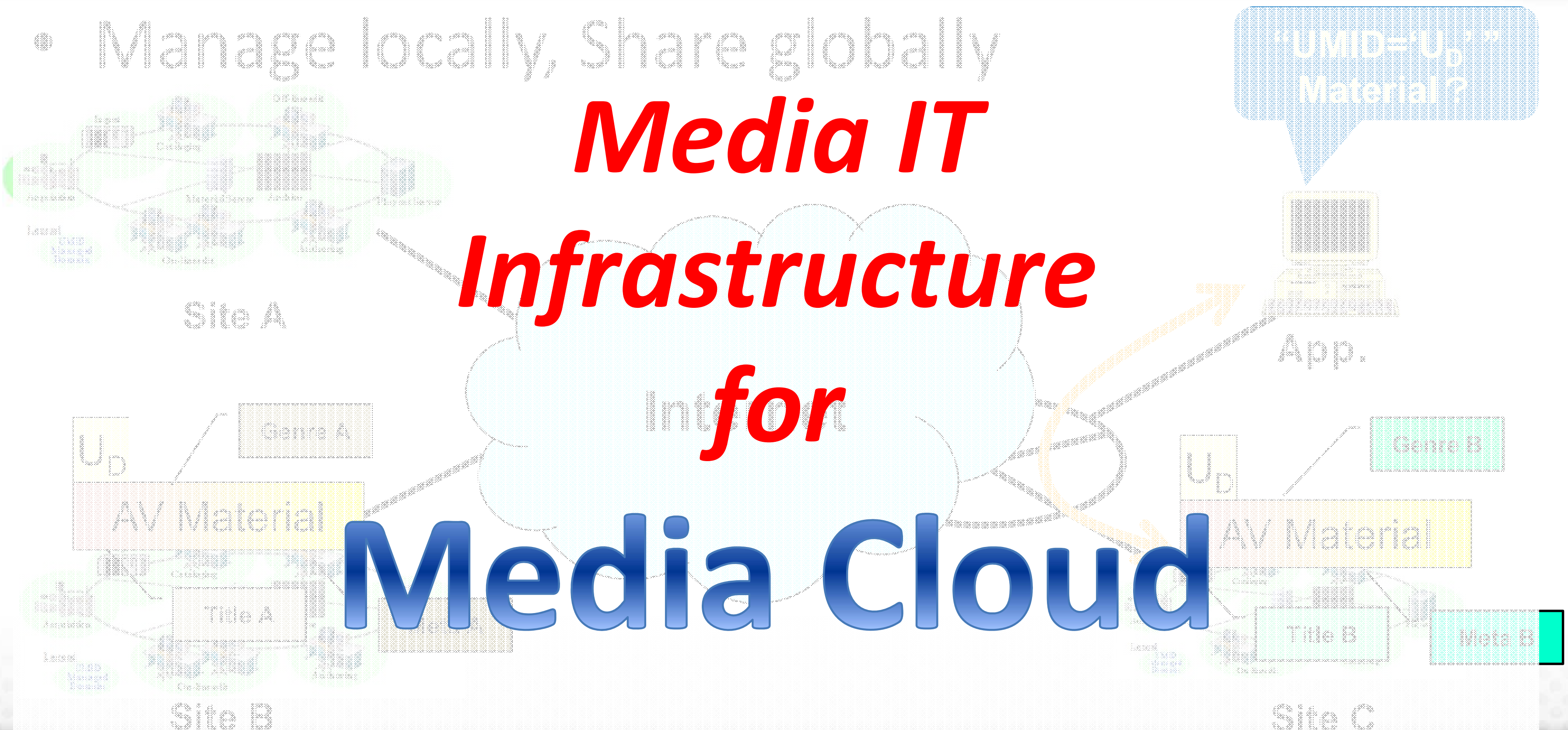
Global AV Material Management

- Manage locally, Share globally



Global AV Material Management

- Manage locally, Share globally



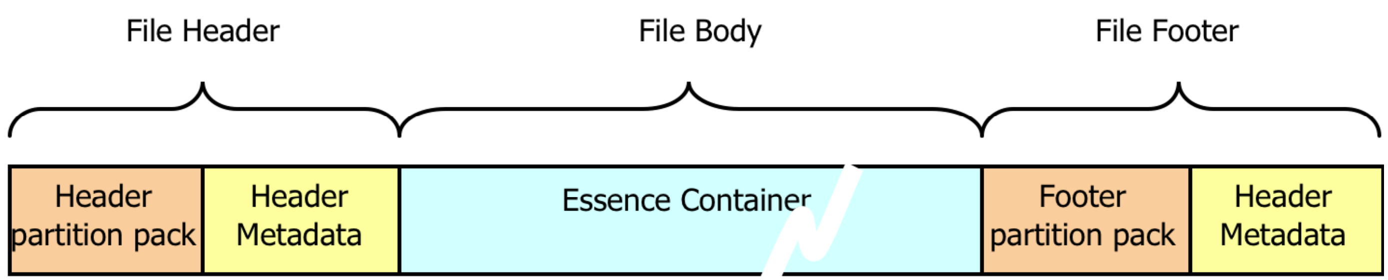
Media IT Infrastructure for

Media Cloud

UMID Applications in MXF

MXF Overview

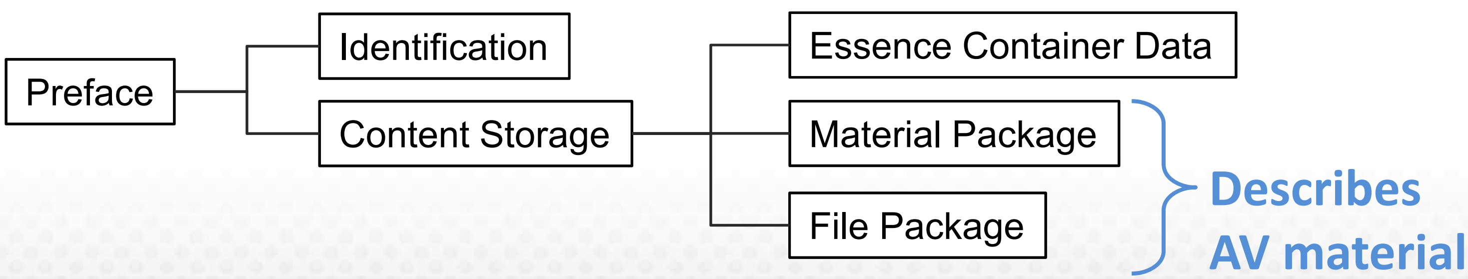
- Material Exchange Format
  - Specifies in SMPTE ST 377-1:2009, etc.
    - Physically: Concatenated KLV(Key-Length-Value) packets
    - Logically: File Header, File Body and File Footer



MXF Header Metadata

- Two kinds of metadata for Header Metadata
  - Structural Metadata
  - Descriptive Metadata

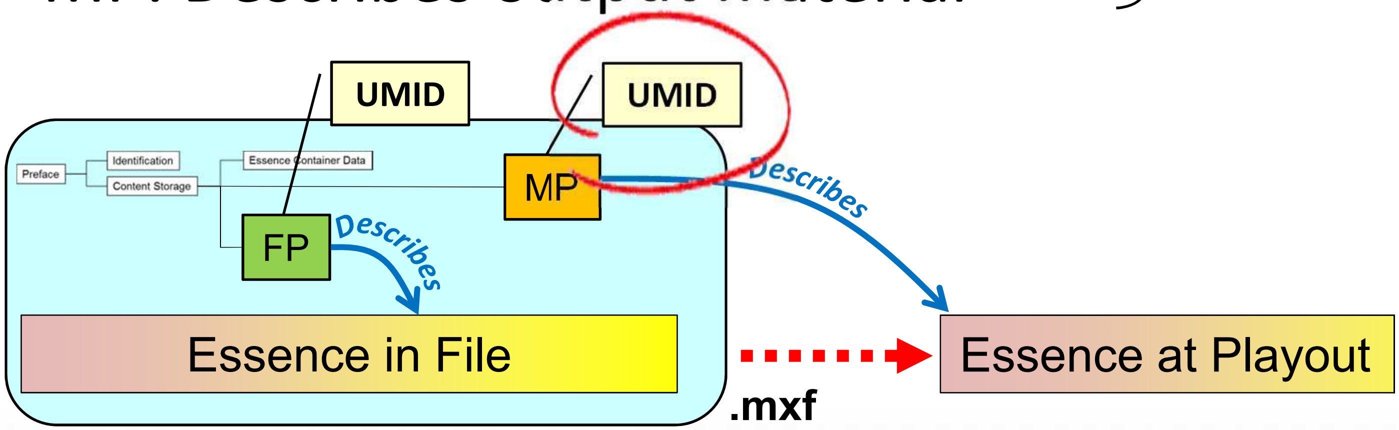
- Structural Metadata tree



MXF Property and UMID

- Material Package (MP) and File Package (FP)
  - FP: Describes material within a file
  - MP: Describes output material

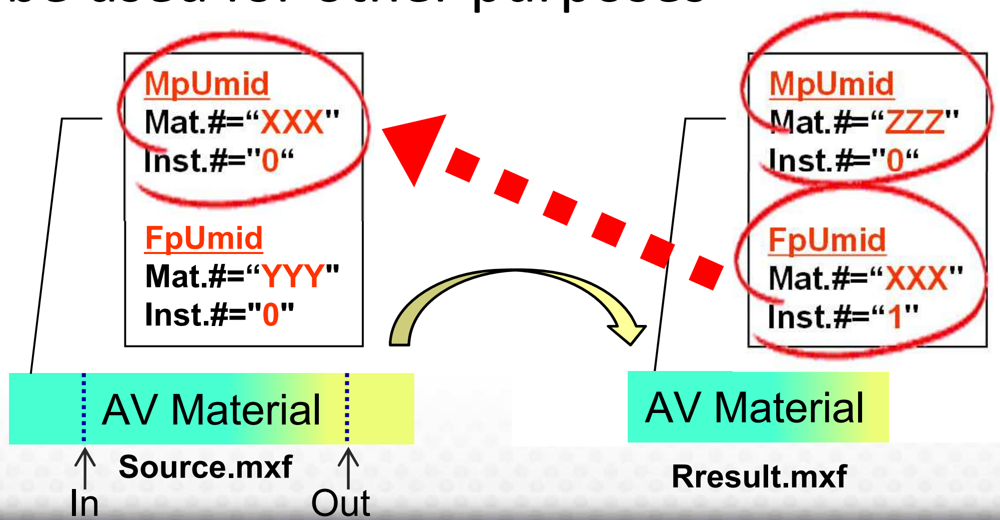
Not always the same



UMID@MP must be a unique material ID !

UMID Applications in MXF

- Material Package UID (**MpUmid**)
  - Used to uniquely identify an AV material (MXF file)
- File Package UID (**FpUmid**)
  - May be used for other purposes



UMID Resolution Protocols

UMID Resolution Protocols

- Resolves a given UMID of AV material to its URL
  - With basic technical metadata
    - Wrapper kind, codec kind, frame size/rate, etc.

- Two plausible UMID resolution protocols
  - DNS (Domain Naming System) – like protocol
  - Standard operation in Web service interface

Conclusions

- **UMID can be much more useful !**
  - Under appropriate common rules for its applications
- **MXF adopts UMID as its core component**
  - To be the media IT infrastructure for the file-based media system
- **Needs standards to achieve it !**
  - UMID Resolution Protocols
  - UMID Application Principles

Author(s)

First Name	Middle Name	Surname	Role	Email	SMPTE Member?
Yoshiaki		Shibata	Consultant	yoshi.shibata@metafrontier.jp	Yes

Affiliation

Organization	Address	Country
metaFrontier dot jp	Yokohama, Kanagawa	JAPAN

Author(s)

First Name	Middle Name	Surname	Role	Email	SMPTE Member?
Jim	Hedley	Wilkinson	Consultant	jim@wilkinson.org	Yes

Affiliation

Organization	Address	Country
Wellspring Digital	Chillaton, Devon	UK

Publication Information

Pub ID	Pub Date
2011 Fall Conference Paper	October 25, 2011

SMPTE Meeting Presentation

UMID Applications in Practices

Yoshiaki Shibata

metaFrontier.jp, Yokohama JAPAN, yoshi.shibata@metafrontier.jp

Jim Wilkinson

Wellspring Digital, Chillaton, Devon, UK, jim@wilkinson.org

Written for presentation at the

[SMPTE 2011 Annual Technical Conference and Exhibition]

Abstract. *UMID is a SMPTE standard identifier that globally uniquely identifies an AV material. Because it is a core component of MXF and AAF, it has been also handled by the products claiming the MXF/AAF support. However, its originally intended use as a globally unique identifier to link AV material to its metadata has been seldom seen in practice.*

This paper aims to achieve its original intention by introducing the concept "UMID Managed Domain" where all AV materials are fully managed via their UMIDs, resulting in any AV material to be unambiguously retrieved by its UMID.

Another important aspect of the UMID Managed Domain is that the domains from various products can be merged to produce a wider domain covering the entire system. To achieve this, however, the UMID resolution protocol spoken among those products needs to be standardized, for which a couple of basic proposals are presented for further discussions.

Keywords. UMID, Material Number, Instance Number, UMID Application Principles, UMID Managed Domain, MXF, AAF, Material Package UID, File Package UID. UMID Resolution Protocol

Introduction

The UMID (Unique Material IDentifier) is a globally unique identifier for audiovisual (AV) material standardized in SMPTE ST 330[1] and its application in RP 205[2]. More than a decade has passed since its initial standardization and it has now become a core component of the MXF (Material Exchange Format) and AAF (Advanced Authoring Format) technologies. Thus products claiming to support MXF or AAF are, by definition, also supposed to handle UMIDs.

According to SMPTE ST 330:

> *"The UMID provides a method of identification for instances of audiovisual material and thus enables the material to be linked with its associated metadata."*

This implies that when metadata associated with a desired AV material is available, the AV material will be obtained by tracking the link to the material. However, because UMID by nature is a dumb number not intended to tell anything about where the material is, even though the metadata with the UMID is at hand, you cannot access the material without the assistance of external functions to resolve the UMID into information where the material is available.

In the media and entertainment (M&E) industry, on the other hand, AV material management by using its unique identifier is a common practice; AV products handling the AV materials within them assign a new identifier to a material at its introduction to the products. As the identifier is based on a proprietary identification scheme, a mapping among those identifiers is required in order for an AV material to be shared among products from different manufacturers. De Geyter, et al.[3] show a couple of options to achieve this: One is to use the material identifiers in a particular product as a master and to make every other products to store the master identifier into their respective databases. Another is to provide a specialized identifier correlation service within a system, which manages the mappings of various identifiers to keep up-to-date.

But a simple question arises for it; Why don't they use the UMID as a common identifier in the system? Because the UMID is a widely used standard identifier, it is supposed to be commonly used in the products even from different manufacturers and, if so, a system composed of such products would have also treated the UMID as an identifier of the material within a system.

However, the reality is that the originally intended use of UMID as a standard globally unique material identifier has been seldom seen in practice.

In this paper, we propose how to use the UMID to achieve its original intention. After the UMID Application Principles are defined, a concept of "UMID Managed Domain" is introduced to embody the principles, including its applicability not only within a product but also to an entire systems and a distributed environment over the internet. Then the UMID application in MXF is specifically discussed based on the principle. Due to the characteristic of MXF, the usefulness of UMIDs in an MXF file as more than just a unique identifier is demonstrated. The requirements and plausible solutions for the UMID resolution protocol for further discussions follow.

UMID Overview

Basic and Extended UMIDs

The UMID is a byte string of either 32 or 64 bytes. The shorter UMID is called the Basic UMID while the longer the Extended UMID, which is composed of a 32 byte Basic UMID followed by the 32 byte Source Pack to store the information on *when/where/who* originally creates the AV material. The Basic UMID is used to globally uniquely identify an AV material as a whole while the Extended UMID to identify it with finer granularity such as a frame.

While an application of the Extended UMID and/or the Source Pack is an interesting topic, we will not go into the details here in this paper. More information for them is available elsewhere[4].

Basic UMID Format

The Basic UMID is a single 32-byte entity composed of four fields as shown in Figure 1 below:

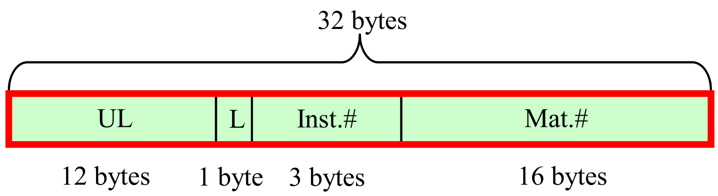


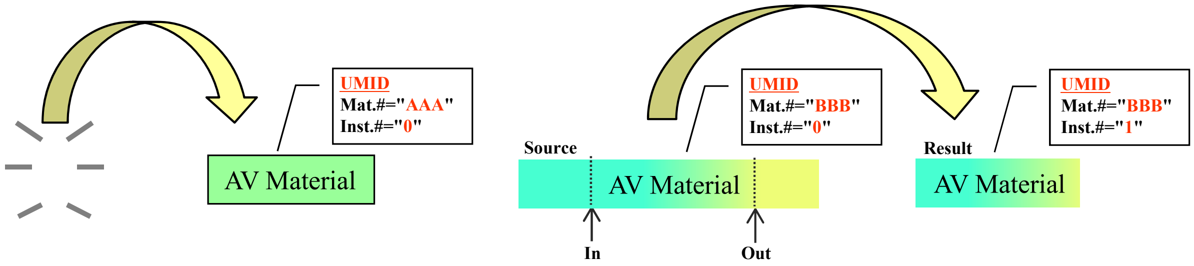
Figure 1. Basic UMID format

- **SMPTE Universal Label** (UL): The first 12 bytes of the Basic UMID constitute the SMPTE Universal Label. The first 10 bytes are fixed values based on the registered ISO label administered by SMPTE[5]. The 11th and 12th bytes indicate the type of material this UMID identifies and the creation methods of values in the fields that follow this label, respectively. SMPTE Universal Labels are 16 bytes long, but the last 4 bytes of the UMID UL are zero and can therefore be omitted when used in the UMID.
- **Length** (L): This 1 byte field specifies the length of byte string that follows. Since 19 bytes follow in the case of Basic UMID, this field is fixed to 13_h.
- **Instance Number** (Inst.#): This 3 byte field specifies whether the Material Number, the field that immediately follows this, is a newly created value or the inherited one from another UMID already existing elsewhere. For the newly created Basic UMID at the introduction of a new AV material, this field shall be fixed to 00_h 00_h 00_h. As a result, if a given UMID is revealed to have its Instance Number being zero-filled, the UMID may be regarded as identifying the original material with a newly created value.
- **Material Number** (Mat.#): This 16 byte field accommodates a globally unique value, which makes the UMID a globally unique identifier. Several creation methods of the value for this field are specified in SMPTE ST 330[1]. An example is given by the combination of the network node number of a device that creates an AV material together with the time snap at which the material is created. Since any single network device (with a single network port) can be globally uniquely identified by its network node number, an AV material with such a UMID will be also globally uniquely identified under the assumption that only one material is to be created at a certain time snap by the device.

Two Distinct Uses of UMID

As originally intended, the primary use of UMID is a globally unique AV material identifier. For example, when a new AV material is created from scratch by acquisition, a UMID is issued and attached to the material automatically so that it can be globally uniquely identified. Note that in this UMID use, the Instance Number (Inst.#) of the UMID is zero-filled as show in Figure 2 (a).

Another use of UMID is a linking tool. Suppose a new AV material is created from an existing source material by partial retrieval. Then a UMID with its Material Number (Mat.#) being inherited from that of the source material while its Instance Number set to non-zero will be attached as shown in Figure 2 (b). In this case, the resulting “truncated” material is associated with the source material via its Material Number, *i.e.*, the UMID of the source material may be easily obtained by just masking the Instance Number field of the resulting material to zero.



(a) UMID as an unique identifier (b) UMID as a linking tool

Figure 2. Two distinct uses of UMID

It should be noted that while there are two uses of UMID introduced above, they are completely exclusive. If a UMID is used as a linking tool as shown in Figure 2 (b), its globally uniqueness cannot be guaranteed because the value space of the 3 byte Instance Number is far from sufficient to accommodate a globally unique value. While it would be possible for an application to carefully control the Instance Number so that the UMID remains unique, the uniqueness is guaranteed only within a predefined closed domain the application can control, *i.e.*, though there is a small chance, it could happen that two independent applications create the truncated materials of different durations but assign UMID whose Instance Numbers are same by chance.

Consequently, when an AV material with UMID of non-zero Instance Number is at hand, the material cannot be managed by the UMID because its globally uniqueness is not guaranteed. If the material is also desired to be managed in a global sense, the UMID needs to be replaced with a newly created value with zero Instance Number (thus as a globally unique identifier).

UMID Application as a Globally Unique Identifier

Goal of UMID Application as an Identifier

While the primary use of UMID is a globally unique identifier for any given item of AV material, the creation and attachment of a unique value to the material is just its first step. Another step to be complemented is that the desired material is made accessible when its identifier is known.

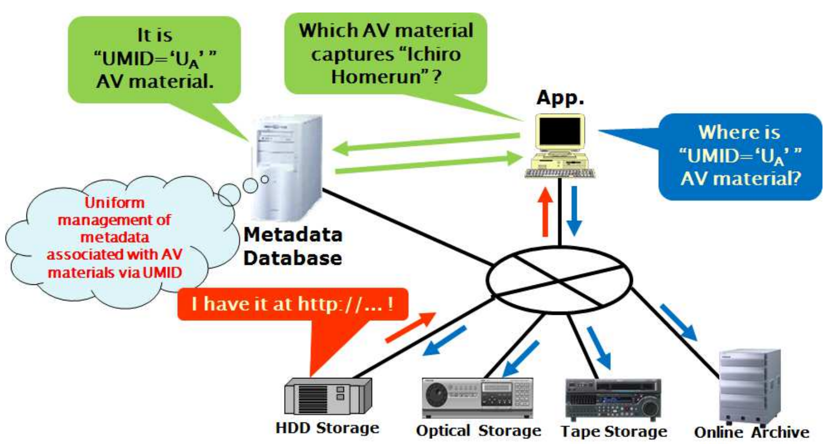


Figure 3. UMID based AV material search

A typical example of such a scenario is the UMID based AV material search as demonstrated in Figure 3. In this scenario, AV materials are stored in various kinds of storage devices connected to the network and metadata associated with an AV material via its UMID is collected and separately stored altogether in the dedicated metadata database for their uniform management.

When an external application desires to obtain an AV material that captures a “Ichiro Homerun” shot for example, then it gives the request to the metadata database accordingly. The metadata database then replies to the application with the requested AV material by its UMID.

Because the UMID by itself cannot tell anything about where to access the AV material, the application needs to resolve the UMID, *i.e.*, to translate the UMID into its corresponding URL (Uniform Resource Locator). In this scenario, the application distributes a query asking “Where is ‘UMID=U_A’ AV material?” and the HDD Storage responds to it with the URL for the requested AV material, which is then used for the application to actually access the material.

Because the application would have difficulty if it has to handle a specific query and response to the UMID resolution request depending on a particular storage device, the UMID resolution protocol needs to be standardized so that any applications can talk to any storage devices in a consistent fashion for that purpose.

UMID Application Principles

UMID Integrity

Behind a scenario described above, each storage device is assumed to appropriately manages its own AV materials with their valid UMID. While several designs would achieve an appropriate UMID based material management, it is worthwhile to identify the UMID Application Principles, the most fundamental rules all products must strictly follow regardless of their implementations, in order to realize the standardized UMID based material management.

First comes is the UMID Integrity, that is expressed as:

Different AV materials shall be identified by different UMIDs.

This should be obvious because otherwise a UMID cannot be a useful material identifier. With this principle, however, it is important to understand precisely what a valid UMID is.

The UMID integrity cannot be determined only by its value. Some UMID value might be syntactically wrong, *e.g.*, starting with bytes other than 06_h $0A_h$ (which may be easily detected by a certain syntax checker). But even if it is syntactically correct, there are some cases where the UMID is not valid. For example, there might exist another AV material having the same UMID but with a different duration. This could happen in a situation where a clone material is first created and then truncated by cutting off the last five minutes of the material.

So a question arises: How can we obtain a material with a valid UMID? There are only two methods to do this. One is to create a new UMID by yourself (by using properly implemented UMID creation function) and to attach it to the material (if not) or to replace the existing UMID with it (if already attached). This is because when you create UMID by yourself, the SMPTE ST 330[1] based UMID creation method ensures that you will have a new globally unique UMID value.

Another method is to obtain the AV material from a UMID Managed Domain, where the integrity of UMIDs are always guaranteed. This ensures that AV materials in the domain are carefully managed based on their valid UMIDs. Further discussion on the UMID Managed Domain will be conducted in the next section.

UMID identification

Another rule that should be made as a part of the UMID Application Principles is regarding what the UMID precisely identifies. This is equivalent to the strict definition of “AV material” used in this paper so far, or the precise meanings of “identical” or “different” for plural AV materials.

To answer the question, the most important requirement to be fulfilled is a clear boundary between “identical” and “different” for everyone without ambiguity. In this sense, “video codec kinds” for example cannot be an answer for the question because their classification boundaries are often obscure, *i.e.*, one might want to distinguish between the H.264/MPEG-4 AVC and the MPEG-2 codecs while another to distinguish the VBR and CBR within the MPEG-2.

Base on those considerations, we propose the answer to the question as:

> *What the UMID identifies is the essence representation at its playout.*

With this principle, one can judge whether two given AV materials are identical or not without ambiguity. For example, if they are clones except for their titles, they are identical. Furthermore, even if they have essences compressed in different kinds but mathematically lossless video codecs, they are identical because the baseband signals at their playout are identical.

On the other hand, if they have essences compressed in different kinds but “visually lossless” video codecs, they are different because even though no visual distinction is detected between them by human eyes, they are mathematically different baseband signals at playout.

Note that this rule does not exclude attempts to assign different UMIDs to materials sharing essences but with different titles, codecs, and so on. Furthermore, together with the first rule in previous subsection, even the assignment of different UMIDs to clone materials is not ruled out. In fact, the UMID Application Principles we propose here define one point of reference as the bottom-line, which we call the Logical Approach. Another point of reference as the top-line, called the Physical Approach, requires all the physical entities to be distinguished by different UMIDs, by assigning new UMIDs even to clone materials.

It is interesting to point out that any sorts of UMID based material distinction in fact exists between those two extreme cases; the Logical and the Physical Approaches. Because the Logical Approach constitutes the bottom-line of such a material identification scheme, it is suitable for the specification as a technology standard that all products can and must follow.

UMID Managed Domain

What is the UMID Managed Domain ?

Based on the UMID Application Principles proposed above, we introduce a concept called UMID Managed Domain, which is composed of AV materials appropriately managed according to their UMIDs. In this domain, all UMIDs used to manage their attached materials are treated as valid globally unique identifiers. In addition, the latest locations of the AV materials are always maintained together with their respective UMIDs so that any of them can be resolved to their respective locations (typically as URLs) for a given UMID without ambiguity.

Figure 4 schematically illustrates the concept of the UMID Managed Domain. In this figure, the Materials 1 to 4 are recorded on a certain physical medium while the Materials 5 and 6 are stored elsewhere. There is a Material Manager (Material Mgr) that manages the AV materials according to their UMID values. The red arrows from the manager to the material denote that the manager recognizes the materials via their UMIDs and their locations (Materials 1, 2, 4 and 5), and can therefore provide a URL for any of those materials in return for a given UMID.

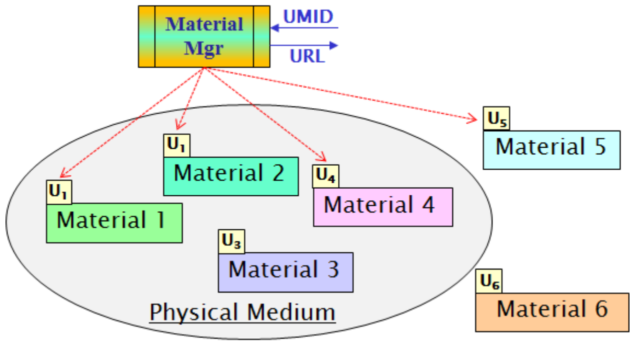


Figure 4. UMID Managed Domain

Note that because the UMID Managed Domain is composed only of materials managed by the Material Manager via their UMIDs, materials not recognized by the manager are out of scope of the domain regardless of their physical locations. For example, even though Material 3 is recorded on the same physical medium as those in the domain, it is not in the domain because it is not recognized by the manager, resulting in the material not to be seen from an external application requesting the Material Manager for its UMID resolution.

Requirements of Material Manager in the UMID Managed Domain

The goal of the UMID Managed Domain is to provide an embodiment of the UMID Application Principles. For the file-based operations, the media files in the domain are properly managed via their respective UMIDs and made accessible via their URLs resolved from given UMIDs.

As shown in Figure 4, It is the Material Manager which plays a key role to accomplish the UMID Managed Domain. It is the Material Manager's responsibility for the UMIDs in the domain to be always valid. In addition, because of the UMID resolution to be realized, a database for a pair of the UMID and its corresponding URL for each media file (called UMID Managed List, hereafter) must be maintained at its latest state. As a result, the Material Manager needs to be involved in most kinds of operations of media files.

In the following, the expected behaviors of the Material Manager for a couple of typical media file operations are discussed.

<u>Media file import into the domain</u>

When a new media file is imported to the domain by copying or moving from elsewhere, the basic behavior of the Material Manager is to attach a newly created UMID (if not) or to replace the existing UMID with a new UMID value (if already attached). Therefore, the Material Manager must detect the UMID of an incoming media file, attach or replace it with a newly created value, and register the pair of the UMID and the resulted file's URL to the UMID Managed List.

An exceptional behavior may be applied when the original location of a media file is also known to be a UMID Managed Domain. In this case, because the UMID attached to the incoming media file is valid, the UMID may be reused when the media file is imported into the domain exactly "as is" (as a clone). Therefore the Material Manager in this case may just detect the UMID of an incoming media file, and register the pair of the UMID and the resulted file's URL in the domain to the UMID Managed List after the completion of the copying or moving operation.

<u>Media file modification at its essence within the domain</u>

When an existing media file is modified by, for example, insert editing, the Material Manager must find the UMID of the media file, replace it with a newly created value, and update the UMID value in the pair of the UMID and the file's URL existing in the UMID Managed List.

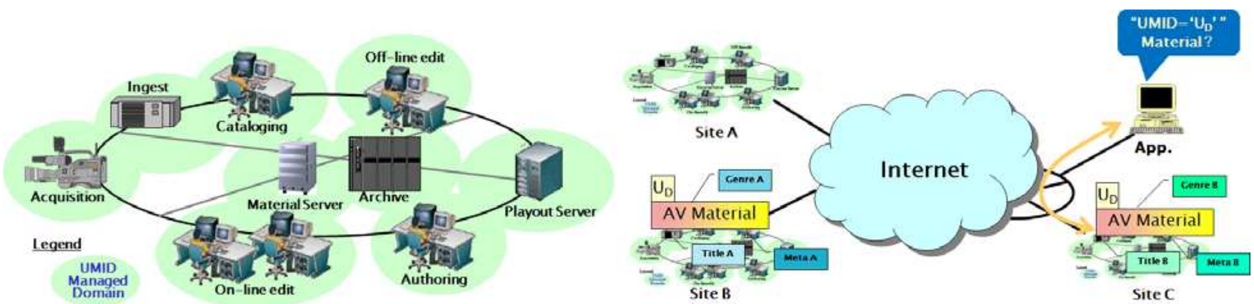
This must be conducted even when only one media file exists for a given UMID in the domain because of a clone file that might exist elsewhere in the world.

Extension of UMID Managed Domain

When a media file is imported to or exported from another UMID Managed Domain, the original UMID attached to the source media file may be used as is for the destination media file. This is regarded as an equivalence of the copy operation of a media file within a single UMID Managed Domain while it is conducted over the boundary of the domains.

This implies that, with a help of the UMID resolution protocol spoken among the Material Mangers for each domain, the plural UMID Managed Domains can be merged to form a single UMID Managed Domain, where the AV materials in the merged domain are uniformly managed in the same way as for those in the original individual domains by using the same UMIDs.

This leads to the realization of so called best-of-breed media system composed of products from multiple manufacturers. As shown in Figure 5 (a), assuming all products participating in a media system support the UMID Managed Domain appropriately, their UMID Managed Domains are merged to cover the entire system, *i.e.*, the media system using the UMID as a common material identifier can be achieved.



(a) Within a system (b) Among multiple sites over Internet

Figure 5. Extension of UMID Managed Domain

Furthermore, thanks to the global uniqueness of the UMID, the UMID Managed Domain can be further extended in a global fashion. Figure 5 (b) shows one of such cases, where AV materials are managed locally at each relevant broadcaster's site (Site A, B and C) but shared globally among them. Because the UMID does not depend on a specific implementation of the AV material management such as for the MAM System used in each site, AV materials to share the same UMID (U_D for Figure 5 (b)) are guaranteed to bring the same essence representation at their playout. Therefore an external application can selectively access the most convenient one among them as shown in Figure 5 (b), where an application accesses the AV material of UMID U_D managed in Site C because of its closer location in the network.

UMID Applications in MXF

MXF Overview

What is the MXF ?

The MXF is a container format standardized in SMPTE ST 377-1[6] and its accompanying document suite. MXF is a container in the sense that MXF contains any kinds and any forms of audiovisual essences in it. MXF also carries metadata, which is to be associated with the whole of, or particular points of the audiovisual essence it contains.

An MXF file, in its physical representation, is composed of a sequence of SMPTE KLV (Key-Length-Value) coded[7] chunks of data (called KLV packet hereafter), which logically form a File Header, a File Footer, and a File Body placed between them, as shown in Figure 6 below.

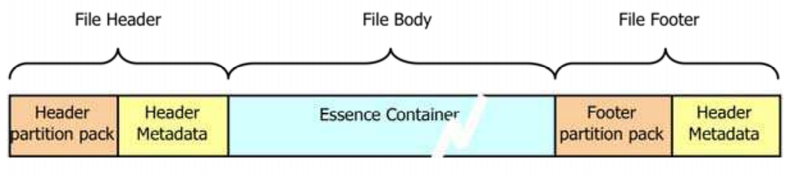


Figure 6. Simple MXF file[6]

In a typical MXF file, the essence data are individually KLV wrapped to form the Essence Container, and stored in the File Body. The granularity of the essence to be KLV wrapped varies depending on the type of MXF file, *i.e.*, from per frame to an entire essence.

Metadata associated with the whole or parts of the file is called the Header Metadata and is contained in the File Header and additionally in the File Footer if desired. Two specific kinds of Header Metadata are defined in MXF; the Structural Metadata and the Descriptive Metadata.

Structural Metadata describes different essence types and their relationship along a timeline. The Structural Metadata is mandatory because it specifies how to play out the essence including its picture size, frame rate, and so on. Because the essence contained in the File Body is typically pure video and/or audio digital data without temporal information, there is no way for the essence to be played out without the Structural Metadata. Consequently, the modification of Structure Metadata may give a strong impact on how the essence is to be played out.

Descriptive Metadata provides information mainly for human use such as titles, synopsis, and so on. While it is important for a user to search a desired MXF file efficiently, we don't go into more details of it in this paper because it does not affect the playout of the essence at all.

The Header Metadata itself is also a sequence of the KLV packets called Sets but they logically form a tree structure by connecting them via strong references (which connect Sets together). Figure 7 shows a simplified logical structure of MXF Header Metadata (the Structural Metadata).

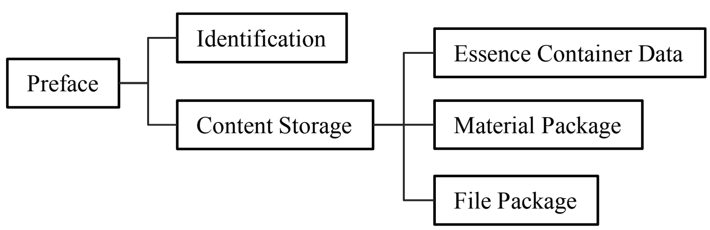


Figure 7. Logical structure of MXF Header Metadata

In this figure, the Preface Set as a root of the Header Metadata signals the kind of MXF file as well as the types of Essences Container and their essence kinds contained in the file.

Below the Preface Set come the Identification and the Content Storage Sets. The Identification specifies MXF file version information, date of creation, etc., and the Content Storage contains three fundamental metadata items, the Essence Container Data, the Material Package and the File Package. The Essence Container Data describes individual Essence Container in the File Body. The Material and the File Packages describe an output timeline of the MXF file and the essence stored in an Essence Container, respectively, which constitute one of the most characteristic parts of MXF and also are vital for the UMID applications in MXF.

MXF with Dual Structure

Among other valuable characteristics of MXF, its dual structure would be one of the most unique characteristics of MXF. In short, what is to be played out is not always the same as what is contained in the file.

Because of its dual structure, an MXF file has two kinds of metadata describing the essence data; the File Package and the Material Package (See Figure 7). Specifically, the File Package is used to describe an Essence Container (and so the essence contained in it) and therefore the same number of instances of the File Package as that of the Essence Containers exist in the file. The Material Package is used to describe the essence to be played out. In other words, it is the Material Package which specifies what we will observe when an MXF file is played out.

Because of such roles of the Material and File Packages, the following features are observed:

- The File Package represents the temporal information of an Essence Container (input timeline) and its technical properties such as codec, frame rate, and so on, by using the File Descriptor,
- The Material Package represents the temporal information of an essence to be played out (output timeline) but does not include the technical properties,
- The Material Package specifies how to play out the essence(s) the File Package(s) describes by referencing to the File Package(s) with its Begin and End timecodes.

These may be schematically modeled as shown in Figure 8.

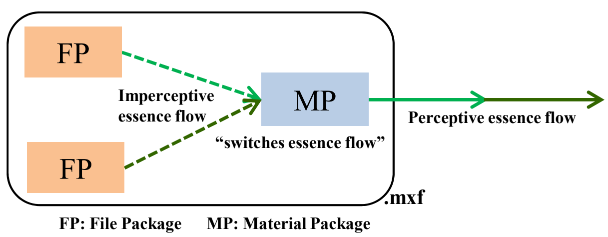


Figure 8. MXF internal model at playout

Based on this figure, the internal behavior at the playout of an MXF file is described as follows:

- Each File Package provides an *imperceptive* baseband essence flow to the Material Package.
- The Material package then controls the essence flow along the output timeline and makes them *perceptive*, resulting in the essence representation we observe at its playout.

The reason why the technical properties of any essence are omitted for the Material Package may be understood that it is the standard baseband essence flows which each File Package generates and supplies to the Material Package, and the Material Package just receives and switches them at playout.

UMID Applications in MXF

UMIDs in MXF

According to SMPTE ST 377-1[6], the Basic UMID is used as a unique identifier (UID) of a Package that describes the essence on a timeline. In the following discussion, let MpUmid and FpUmid denote the Package UIDs of Material and File Packages, respectively.

Applications of Material Package UID (MpUmid)

While an originally expected role of the MpUmid would be a unique identifier of an instance of the Material Package, it is also regarded to uniquely identify the essence on an output timeline, *i.e.*, it uniquely identifies the essence representation at its playout. Hence, based on the UMID Application Principles, the MpUmid is regarded as a globally unique identifier of an MXF file.

Consequently, the UMID Managed Domain composed of MXF files with their MpUmid needs to be established. The Material Manager in the domain is expected to behave in the way as previously discussed. For example, when an MXF file is imported from somewhere other than the UMID Managed Domain, the Material Manager must detect the MpUmid of an incoming MXF file, replace it with a newly created value, and then register the pair of the MpUmid and the resulted file's URL to the UMID Managed List.

As for the MXF file modification at its essence, the Material Manager must detect the MpUmid of the modified MXF file, replace it with a newly created value, and update the pair of the MpUmid and the file's URL in the UMID Managed List accordingly. Since it is the Material Package which controls the essence to be played out, the modification of the Material Package often requires its MpUmid being updated except that it does not affect the essence to be played out at all.

Applications of File Package UID (FpUmid)

The FpUmid is a unique identifier of an instance of the File Package and so can be regarded to uniquely identify the essence flow from its File Package to the Material Package for its playout. Because this is a conceptual essence flow effective only within a MXF file, there is *theoretically* no way for an external application to access it even when a given UMID is resolved to an MXF file containing the File Package with the UMID value as FpUmid. Hence, the usefulness of managing the FpUmid would be quite limited in general.

On the other hand, the uniqueness of FpUmid is crucial within relevant MXF file(s) because its File Package is uniquely referenced by the Material Package within the file(s).

Based on these considerations, though a default treatment of the FpUmid would be to assign a newly created value for it, there is another application for the FpUmid to be made more useful in practical situations if the scope of the FpUmid uniqueness to be within MXF file(s) is accepted.

An example of such an FpUmid use is given by the partial retrieval of an MXF file. Suppose a resulting MXF file "Result.mxf" is obtained by partially retrieving a source MXF file "Source.mxf" with its In/Out points in the way as shown in Figure 9.

Because "Result.mxf" is a newly created MXF file, its MpUmid must be the newly created value with its zero Instance Number (Inst.#). On the other hand, if its FpUmid is permitted not to be globally unique, another use of UMID as a linking tool can apply, *i.e.*, because the essence described by the File Package is created from the output of "Source.mxf", it is reasonable for the FpUmid to be also created from the MpUmid of "Source.mxf". Specifically, the Material Number

(Mat.#) of the FpUmid for “Result.mxf” is inherited from that of the MpUmid for “Source.mxf”, while its Instance Number be non-zero value to indicate its use as a linking tool.

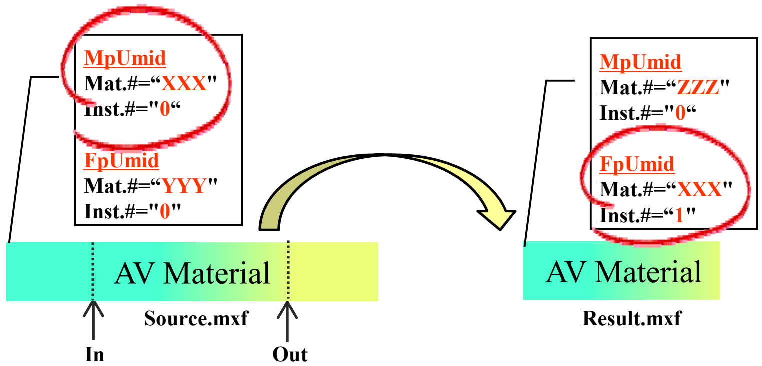


Figure 9. UMID application in partial retrieval of an MXF file

Note that since an application executing such an operation can fully manage the creation of all UMIDs to be contained in the resulting MXF file(s), it is easy for the application to assign UMIDs that are unique within MXF file(s) by just controlling the non-zero Instance Number.

This kind of FpUmid treatment brings additional usefulness of the FpUmid. While it is obvious in the figure, the resolution of the FpUmid of resulting “Result.mxf” with its Instance Number masked to zero will lead to the source MXF file from which the resulting MXF file is created.

UMID Resolution Protocol

Requirements for UMID Resolution Protocol

While AV materials can be fully managed by using their respective UMID values, it is useless if these materials cannot be accessed from external applications. It is also obvious that the method to access material via the UMID must be standardized in order for an application to access any materials managed by any products in a consistent fashion.

While the detailed specifications of the UMID resolution protocol is still under study, we would suggest its requirements and plausible solutions in this subsection.

The requirements identified so far include:

- UMID must be resolved to the URL of its corresponding media file,
- Some technical metadata that hints the requester to judge its playability should be provided because the UMID value by itself contains no information as to whether it is playable or not,

DNS-like Protocol

One possible solution would be to borrow the idea of the Domain Naming System or DNS, where each 4 byte IP address is resolved to its associated URL and vice versa. In this approach, the following items need to be considered:

- Dedicated port number to be registered to IANA,
- Payload as a binary format for request and response,
- RP 224 (SMPTE Label)[8] based technical metadata.

Standard Operation in Web Service Interface

With the SOA (Service Oriented Architecture) based technology becoming popular in the M&E industry[9], it is convenient to specify the Web service version of the UMID resolution protocol. In this approach, the following items need to be considered:

- Independency of underlying protocol,
- XML Schema for messages to be exchanged and its Message Exchange Pattern (MEP),
- Both SOAP and REST to be taken into account,

Conclusion

The reason why the originally intended use of UMID as a standard globally unique material identifier has been seldom seen in practice is now revealed by a lack of fundamental rules for its application, including the UMID resolution protocol. While the UMID application has been described in SMPTE RP 205[2], we cannot but say that its content is insufficient when we observe UMID uses in reality. We believe this work should be a contribution for the RP document to be updated to improve its scope and quality.

It is fortunate for the MXF to adopt the UMID as its mandatory item. Based on the fundamental rules established in this work, an MXF file is appropriately managed in a global fashion by using its Material Package UID. Furthermore, thanks to the characteristic of MXF, a link to the source material, one of the most frequently requested behaviors, can be successfully implemented.

It is clear that such fundamental rules about the UMID application should be widely disseminated in the industry in order to maximize its usability. Therefore, the standardization activity for the UMID application is to be proposed shortly in order for the UMID together with the MXF to constitute a part of the informational infrastructure for the file-based media systems

References

1 SMPTE Standard ST 330:2011 Television – Unique Material Identifier (UMID).

2 SMPTE Recommendation Practice RP 205:2009 – Application of Unique Material Identifiers in Production and Broadcast Environments.

3 Matthias De Geyter, Nico Oorts, and Luk Overmeire, "Integration Demands on MAM Systems: A Proof of Concept Solution", SMPTE Motion Imaging Journal, November/December 2008, pp.38-46.

4 Yoshiaki Shibata and Jim Wilkinson, "White Paper on UMID and its Applications", the technology brochure distributed at Sony Booth in NAB 2002 (April 2002),
available at: http://metafrontier.jp/drupal/sites/default/files/papers/umid4Nab2002TechBrochure.pdf

5 SMPTE Standard ST 298:2009 – Universal Labels for Unique Identification of Digital Data.

6 SMPTE Standard ST 377-1:2009 Material Exchange Format (MXF) – File Format Specification.

7 SMPTE Standard ST 336:2007 Data Encoding Protocol Using Key-Length-Value.

8 SMPTE Recommendation Practices RP 224v10:2009 SMPTE Labels Register.

9 EBU-AMWA FIMS (Framework for Interoperable Media Service) Task Force,
available at: http://wiki.amwa.tv/ebu/index.php/Main_Page