SMPTE Meeting Presentation

Latest Status of UMID Application Project in SMPTE

Yoshiaki Shibata (Chair, SMPTE TC-30MR SG UMID Applications)

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

Written for presentation at the

SMPTE 2013 Annual Technical Conference & Exhibition

Abstract. *A project is ongoing at SMPTE Standard Community to enhance applications of UMID, the SMPTE standard globally Unique audiovisual Material IDentifier. While already identified the UMID Application Principles, the fundamental rules every UMID-aware product must strictly follow, which are to be reflected to upcoming revised SMPTE RP 205, how to realize the UMID Resolution Protocol to convert a given UMID into its corresponding URL is still under intensive study.*

In this paper, we report the latest status of the project and the feasibility study of DNS (Domain Name System) to be used as a basis of the UMID Resolution Protocol implementations.

Keywords. MXF, UMID Application Principles, UMID Managed Domain, UMID Resolution Protocol, DNS

(The SMPTE disclaimer is on a footer on this page, and will show in Print Preview or Page Layout view.)

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the Society of Motion Picture and Television Engineers (SMPTE), and its printing and distribution does not constitute an endorsement of views which may be expressed. This technical presentation is subject to a formal peer-review process by the SMPTE Board of Editors, upon completion of the conference. Citation of this work should state that it is a SMPTE meeting paper. EXAMPLE: Author's Last Name, Initials. 2011. Title of Presentation, Meeting name and location.: SMPTE. For information about securing permission to reprint or reproduce a technical presentation, please contact SMPTE at jwelch@smpte.org or 914-761-1100 (3 Barker Ave., White Plains, NY 10601).

Introduction

The UMID (Unique Material IDentifier) is a globally unique audiovisual (AV) material identifier standardized by SMPTE as SMPTE ST 330[1] and RP 205[2]. Although UMID has been widely disseminated over the industry as a mandatory component of the MXF (Material eXchange Format) and AAF (Advanced Authoring Format) technologies since their introduction, its originally intended use as a unique material identifier to associate the material with its external metadata is seldom seen in practice.

In 2011, we have pointed out[3] that this is because of lack of additional technologies needed to be industry standardized to realize such UMID applications, such as the UMID Application Principles and the UMID Resolution Protocol. To address this issue, a project[4] (called the UMID Application Project, hereafter) has been established in the SMPTE SC (Standard Community) on April 2012 with the project scope as:

1. To explore the best practice of UMID applications,
2. To identify typical UMID Application Principles and collate the fundamental rules every UMID-aware product needs to adopt,
3. To identify relevant technologies needed to be additionally standardized.

Then, on February 2013, its first study report[5] has been submitted to TC-30MR in SMPTE SC, which recommends the UMID Application Principles to be SMPTE standardized, and another project[6] has started since June 2013 to revise RP 205[2] based on the recommendation.

In this paper, we report the latest status of the UMID Application Project with a brief introduction of the contents of the first study report, including the UMID Application Principles to be defined in the revised RP 205. The feasibility study of DNS to be used as a basis of the UMID Resolution Protocol is also discussed with its preliminary experimental result.

UMID and its Applications

What is the UMID?

The UMID is a byte string of either 32 or 64 bytes. The shorter UMID is called the Basic UMID and the longer the Extended UMID which is composed of the 32-byte Basic UMID followed by the 32-byte Source Pack to store the information on when/where/who originally creates the material. The Basic UMID is primarily used to uniquely identify the 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 focus on the applications of the Basic UMID that is the core component of a unique material identifier.

Figure 1 below shows the Basic UMID format, which is composed of the following four fields:

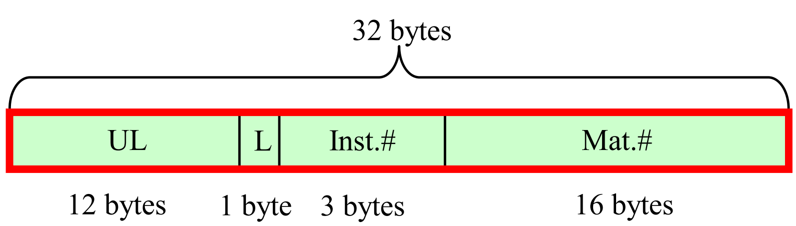


Figure 1: Basic UMID Format

- **SMPTE Universal Label (UL)**: The first 12 bytes of the Basic UMID constitute the SMPTE UL. The first 10 bytes are fixed values based on the registered International Organization for Standardization label administered by SMPTE. The 11th byte indicates the type of material this UMID identifies. The 12th byte is divided into top and bottom nibbles: the top nibble composed of 4 most significant bits (MSBs) and the bottom nibble of 4 least significant bits (LSBs) indicate the number creation methods for the Material Number (Mat.#) and the Instance Number (Inst.#), respectively,

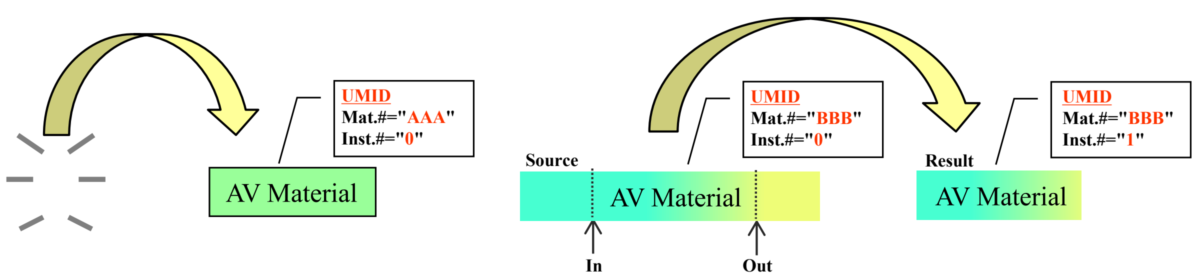
- **Length (L)**: This 1-byte field specifies the length of byte string that follows. Since 19 bytes follow in the case of Basic UMID, this field is fixed to 13_h,

- **Instance Number (Inst.#)**: This 3-byte field specifies whether the Mat.#, the field that immediately follows this, is a newly created value or the inherited one from another UMID already existing elsewhere. For a newly created UMID, this field must be zero-filled (00_h 00_h 00_h), indicating that the Mat.# is a newly created value. While several methods are defined for the creation of non-zero Inst.#, the distinction of zero or non-zero value for it is significant for most UMID applications,

- **Material Number (Mat.#)**: This 16-byte field accommodates a globally unique value, which makes the UMID also a globally unique material identifier. Several creation methods of a value for the field are specified, among which an example is given by a combination of the network node number of a device creating a material together with a time stamp at which the material is created. Because all network node numbers are globally unique, a material with UMID of this combination can be also globally uniquely identified under the assumption of a single material being created at a particular time stamp. Note that this method creates a self-identifying globally unique value without access to either the registration authority or the central database, which is in common use in the information technology system such as for UUID (Universally Unique IDentifier).

Two Distinct Uses of UMID

The primary use of UMID is a globally unique material identifier. When a new AV material is created from scratch by *e.g.*, acquisition, a new UMID is created and attached to the material automatically so that it can be globally uniquely identified by the UMID value. In this case, while the Material Number (Mat.#) for the UMID is a newly created globally unique value, the Instance Number (Inst.#) must be zero-filled as show in Figure 2 (a) below, indicating that it is the original AV material.



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

Figure 2: Two Distinct Uses of UMID

Another use of the UMID is as a linking tool. When a new AV material is created from existing source material by *e.g.*, partial retrieval, a UMID whose Mat.# being inherited from that of the source material while the Inst.# set to non-zero will be attached to the resulting material as shown in Figure 2 (b) above. In this case, the resulting material is logically associated with its source material via their Mat.#, *i.e.*, the UMID of the source material can be easily obtained by just masking the Inst.# of UMID for the resulting material to zero.

While there are two uses of UMID as a globally unique material identifier and as a linking tool, those two uses are completely exclusive. If a UMID is used as a linking tool as is demonstrated in Figure 2 (b), it can no more be a guaranteed globally unique material identifier because of the value space of the Inst.# of only 3-byte long, which is far from sufficient to accommodate a globally unique value.

This leads to the conclusion that when a material to hand has a UMID with a non-zero Inst.#, the material cannot be managed by the UMID in a global sense. In the following, we focus on the UMID use as a globally unique material identifier only, resulting in that the Inst.# field is always zero-filled.

UMID as a Globally Unique Material Identifier

When a UMID is used as a globally unique material identifier, the most important role of the UMID would be to unambiguously associate an AV material with something else, typically, with metadata describing the material. Assuming an MXF file and an XML (eXtensible Markup Language) file are given as a material and its associated metadata, respectively, Figure 3 below schematically illustrates how they are associated via UMID.

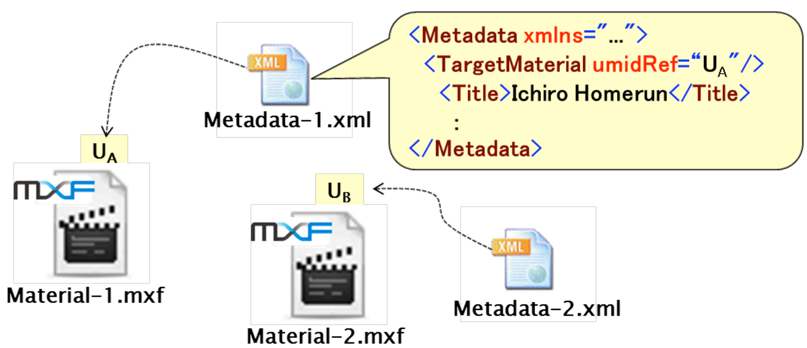


Figure 3: Material Associated with Metadata via UMID

In this figure, the MXF files, “Material-1.mxf” and “Material-2.mxf”, represent certain AV materials which are globally uniquely identified by the UMID U_A and U_B, respectively. According to the MXF file format specification[7], an MXF file has a field that must be filled with a valid 32-byte basic UMID in its header metadata so that the UMID is firmly tied up with the MXF file. As a result, “Material-1.mxf” containing the UMID U_A value at its header metadata is to be globally uniquely identified by the UMID U_A.

When metadata associated with a material is given, it needs to unambiguously specify the material it is associated with, and it is the UMID which works as a *hook* to hang the metadata. In the case of Figure 3, “Metadata-1.xml” describing “Ichiro Homerun” for the title is associated with “Material-1.mxf” via U_A, which is represented by explicitly specifying U_A for the value of the umidRef attribute in the TargetMaterial element.

It should be noted that such an association has been conventionally implemented by using a URL (Uniform Resource Locator) such like “http://server.example.com/materials/Material-1.mxf” so far, which directly indicates where the target material as an MXF file locates. The use of UMID, on the other hand, is regarded as an indirect logical way of the association, which brings a benefit of much higher flexibility than the URL-based one though an additional tool is required for the UMID-based association to work in practice, as is discussed in the following section.

UMID based Material Search and UMID Resolution Protocol

One of the primary roles of metadata is its use for the AV material search. Because of huge difference in data sizes between the material and its associated metadata, it is reasonable and in common practice that the metadata is to be stored separately from the material.

Based on the association between a material and its metadata via UMID, an application scenario, called the UMID based material search, is schematically demonstrated in Figure 4 below, where materials are stored in various kinds of material servers ("Ingest Server", "Near-line Material Server", "Archive", and "Playout Server") connected to the network in a media production system and metadata associated with a material via its UMID is collected and separately stored altogether in the dedicated metadata database ("Metadata Database") for their uniform management.

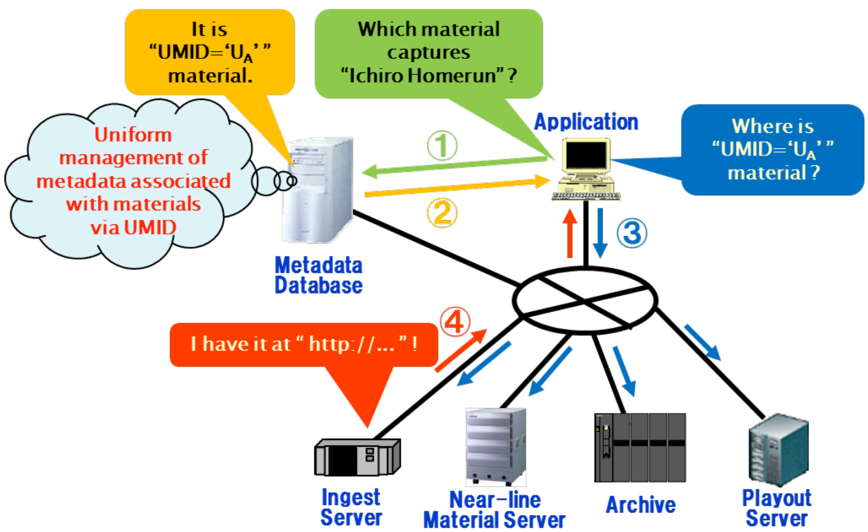


Figure 4: UMID based Material Search

In this scenario, when an external application ("Application"), such as video editing, desires to obtain a material that captures an "Ichiro Homerun" scene, for example, it will give a query to the metadata database accordingly. The metadata database will then reply to the application with the desired material by its UMID.

Because the UMID itself cannot tell anything about where to access the desired material, the application needs to resolve the UMID, *i.e.*, to convert the UMID into its corresponding URL. In this scenario, the application will distribute a query asking "Where is 'UMID=U_A' material?" and the ingest server ("Ingest Server") will respond to it with the URL for the desired AV material, which is then to be used for the application to actually access the material.

It should be noted that, in this application scenario, because each material server would come from different vendor, it is essential for the UMID Resolution Protocol, or the conversation method between the application and the material servers in order to convert a given UMID into its corresponding URL, is industry standardized.

UMID Application Principles

Furthermore, it is assumed in the previous application scenario that all the materials stored in any material servers are appropriately managed based on their UMIDs as globally unique material identifiers. But what does it mean by “appropriately managed”?

To answer this question, it is also essential to clearly define the fundamental rules applicable to all kinds of UMID applications that need to be strictly followed by all UMID-aware products, which is called UMID Application Principles.

For example, if different materials share a single UMID, such a UMID is apparently useless as a unique material identifier at all, implying that a statement such as “*Different Materials shall be globally uniquely identified by different UMIDs*” should be defined as a part of the UMID Application Principles.

It is worthwhile to note that while the above statement looks as a matter of course and easily complied by any product at a glance, it is not so much straightforward than its appearance. For example, while a variation media file would be often created in such a way that a clone of an original media file is initially created and its picture essence is partially overwritten with another picture, the statement requests that a UMID of the variation media file must be different from the original one, which can never happen unless it is intentionally replaced with a newly created UMID value.

So, what happens when it is metadata such as a title or synopsis which is modified within the clone of a media file? While there might be some special cases where those modified media files are still desired to be distinct for some purpose, which themselves should not be ruled out, their distinction is not significant for most cases because they are identical at their playout and thus when they are consumed.

In order for the above consideration to be shared among UMID applications, a statement such as “*If more than one material is uniquely identified by a single UMID, their representations at playout shall be identical bit by bit on the timeline*” should be also defined as a part of the UMID Application Principles, which in fact precisely defines what a UMID uniquely identifies, *i.e.*, it is a bit-stream of a material at its playout.

It should be noted that while the UMID Application Principles themselves look as statements of course, they are not provided as a nonbinding target but stipulated as mandatory rules all the UMID-aware products must strictly follow in order for the UMID to be consistently treated among them. In other words, they need to be industry standardized.

UMID Application Project

TC-30MR SG UMID Applications[4]

To address the issues of UMID application introduced so far, a new project was established in 2012 under the Technical Committee TC-30MR Metadata & Registers in the SMPTE Standard Community (SC), which is known to as the UMID Application Project chaired by the author.

At the time of this writing, there are 47 members from more than 30 major organizations.

The project has initially started as SG (Study Group), named TC-30MR SG UMID Applications. This is because more than a decade has been already passed since the initial introduction of UMID to the industry and, although UMID has been eventually useless in practice, there have been reported some experimental trials[8,9] or implementations in specific products[10].

Therefore the initial task of the project was to collect existing practices of the UMID applications, including organizing the UMID application conference, which was successfully held as a part of the SMPTE engineering block meeting in Geneva 2012, where eight interesting topics on the UMID applications were presented.

After analyzing the collected existing practices of the UMID application, we identified candidate statements for the UMID Application Principles and collated them with the existing practices. We also explored the best practices of the UMID applications via their analysis, which has brought an overview of the UMID treatments in a media production workflow chain.

On February 2013, we have submitted our first study report, the Study Report Part 1[5], to the parent technical committee, TC-30MR, and as recommended by the report, a new activity[6] has started to revise SMPTE RP 205[2] so that it appropriately reflects the study result including the UMID Application Principles.

We've also started to create the second study report that addresses additional technologies needed to be SMPTE standardized to enhance the UMID applications. While it is still under development within the study group, candidate technologies at the time of this writing include the DNS-based UMID Resolution Protocol that is further discussed in the following section, UMID based program package exchange, and the UMID applications specifically for MXF.

Study Report Part 1

In this section, the content of the Study Report Part 1[5] is briefly introduced.

Candidate Statements for the UMID Application Principles

First, the candidate statements for the UMID Application Principles are proposed. Specifically, they are composed of the following seven principles:

- ✓ Principle 1 Definition
- ✓ Principle 2 UMID Creation
- ✓ Principle 3 UMID Integrity
- ✓ Principle 4 UMID Identification
- ✓ Principle 5 UMID Inheritance
- ✓ Principle 6 Extended UMID
- ✓ Principle 7 Source Pack

In fact, the statements proposed to be defined for the UMID Application Principles in the previous section are quoted from the Study Report Part 1, where the former and the latter in the previous section correspond to the Principle 3 (UMID Integrity) and the Principle 4 (UMID Identification), respectively.

It should be noted that because UMID can be used not only as a globally unique material identifier but also as a linking tool as shown in Figure 2, the principle relevant to the latter UMID use is also included as the Principle 5 (UMID Inheritance). Furthermore, because there is the Extended UMID composed of the 32-byte Basic UMID immediately followed by the 32-byte Source Pack, the principles relevant to the Extended UMID need to be included, which is covered by the Principles 6 (Extended UMID) and the Principle 7 (Source Pack).

UMID Managed Domain

Second, the UMID Managed Domain is introduced as an embodiment of the UMID Application Principles. The UMID Managed Domain is a conceptual domain composed of AV materials, each of which is globally uniquely identified by a UMID attached to it.

By definition, UMID attached to a material in the UMID Managed Domain is guaranteed as a valid UMID in the sense of the UMID Application Principles as a globally unique material identifier. Furthermore, because UMID by itself contains no information to access a material as already discussed, it is crucial for the access information such as a URL to be appropriately managed together with its corresponding UMID.

Figure 5 below schematically illustrates the concept of UMID Managed Domain. In this figure, “Materials 1” to “Material 4” are recorded on a certain physical medium as media files while “Materials 5” and “Material 6” are stored elsewhere. In addition, there is a material manager (“Material Manager”) that manages the materials with respect to their UMID and URL correspondences.

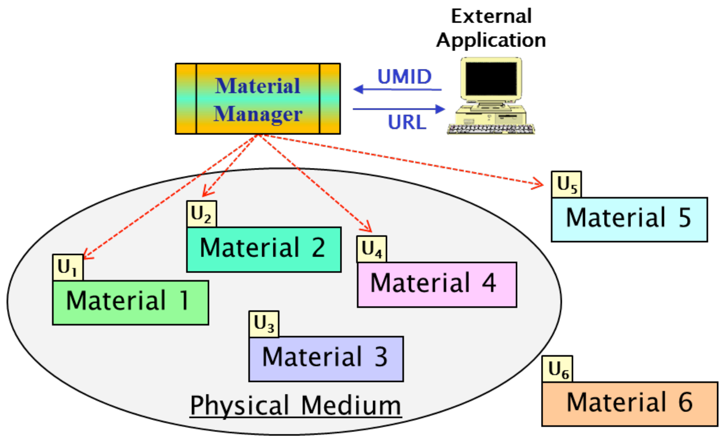


Figure 5: UMID Managed Domain

It should be noted that only the materials managed by the material manager (via their UMIDs) can constitute the UMID Managed Domain. As a result, even materials recorded on the same physical medium such as “Material 3” cannot be a part of the domain when it is not recognized by the material manager, *i.e.*, no red arrow from the manager to the material in Figure 5, while materials recorded elsewhere such as “Material 5” can be a part of it because it is recognized by the material manager.

Note also that, from the viewpoint of an external application (“External Application”), because it is the material manager to which the application gives a query on a desired material (as a media file) via its UMID for its URL as is shown in Figure 5, only the materials recognized by the material manager, *i.e.*, those as parts of the UMID Managed Domain, can be seen and therefore accessible from the application.

It should be emphasized that in order to maintain the validity of all the UMID attached to the materials in the UMID Managed Domain, the material manager should play a crucial role. Furthermore, it is also the manager’s responsibility to keep the latest correspondence between a UMID and its URL in order to address the UMID resolution to its corresponding URL. Consequently, the material manager, maintaining a database of the UMID and URL correspondence for each material as a media file, will be eventually involved in any kinds of material manipulation in the domain.

In the Study Report Part 1, behaviors the material manager should take are discussed in details for various material manipulations such as a material creation, modification, deletion within the domain, and a material import from or its export to the domain.

UMID Application Examples

Third, more than a dozen of the UMID application examples collected during the study is introduced. Although the original intention of the study was to identify the best practices of the UMID applications, there is no clear criterion to determine the best ones. Hence, all the collected examples of the UMID applications based on from the traditional VTR/SDI environment to the modern SOA (Service Oriented Architecture) one are introduced, while an application is generalized by extracting its essential behavior if it is specific to a certain product.

Among those examples, the modern SOA based one is briefly introduced in the following.

As the market is rapidly changing such as with growing competition across a variety of delivery platforms including traditional broadcasting, mobile and the Internet, more attention has been paid to the system that is flexibly configurable to meet a quick change of business requirements such as system scale-up or workflow changes. Thanks to the drastic increase of computational power and hardware resources available to hand, the SOA approach, which used to be only for the enterprise system in early days, has become popular even for the media production system, and relevant standardization activities such as FIMS (Framework for Interoperable Media Services)[11] has taken place with much attention.

In this approach, each component that provides a certain function in a media production system such as ingest, transform and transfer is abstracted as a media service that is independent of a specific workflow, and with the help of so called SOA middleware, those services are dynamically integrated to form an actual media production system at a given workflow information.

Because each service is independent of a specific workflow, a system composed of those services is expected to be flexibly configured to meet a dynamic change of workflow. Furthermore, thanks to little or no interdependency among services, arbitrary services can be freely combined to form a system without unexpected side effects that are common for the reconfiguration of traditional media production system.

Such a characteristic of the SOA approach is called a loose coupling.

In addition, information exchanged among services is also abstracted as an XML instance in a SOA-based media production system; *i.e.*, even for the material treatment, it is an XML instance describing a material as a proxy which is actually exchanged among media services until a real material manipulation is certainly required, resulting in again contributing to a flexible integration of services.

If we apply this concept to the system shown in Figure 4, it is revealed that the system can be divided into two layers: the upper layer containing “Metadata Database” and “Application” in the figure and the lower layer containing various material servers as is depicted in Figure 6.

In Figure 6, the upper layer, called the Application Layer, is the one where the abstracted media services are flexibly integrated to form a media production system for a given workflow. In this layer, although data that goes across the layer is tiny in its data size (typically as an XML instance), it needs to be handled in a very flexible and complex way.

The lower layer, called the Media Layer, is the one where materials are actually manipulated. Although the material as data is usually huge in its data size, the data treatment is simple and straightforward, *i.e.*, data intensive processing such as material creation, material movement or material transformation.

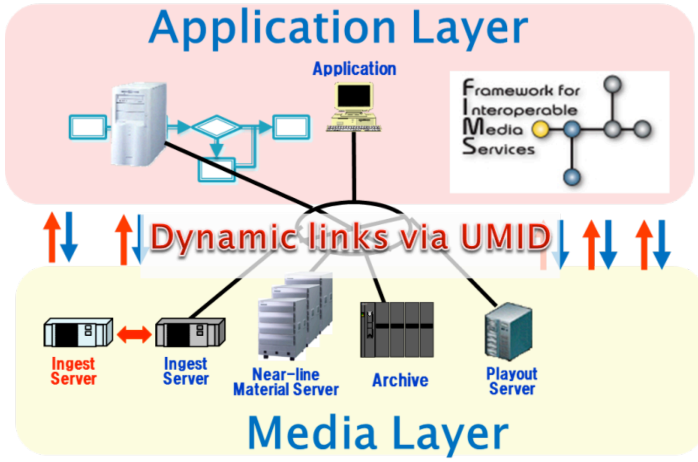


Figure 6: Loose Coupling between Application and Media Layers

While the SOA approach has brought flexibility in a system reconfiguration, it is only within the Application Layer. In other words, even though we want to replace the ingest server ("Ingest Server") with its spare at its failure or to increase the number of near-line material servers ("Near-line Material Server") in order to accommodate an increase of materials, they cannot be supported by the flexibility in the Application Layer because the workflow information itself remains unchanged.

Furthermore, the URL, which is typically used in a conventional system to associate data in the Application Layer such as an XML instance describing a material with that in the Media Layer such as the material as a media file, further deteriorates the flexibility in such treatments of material servers because it requires an explicit representation of hostname that is usually tightly coupled with a particular material server.

The UMID, together with its resolution protocol, will address this issue. By introducing a UMID as a globally unique material identifier, the material related data in the Application Layer is logically associated with its target material in the Media Layer via the UMID. Then, the UMID is resolved to an access method for the target material when actual manipulation of the material is required, which is called the dynamic linking of the Application and the Media Layers via UMID.

This results in easy replacement of the ingest server with its spare (containing the same materials as backup) in Figure 6. Furthermore, because where to store a material in the Media Layer is abstracted as its UMID from the viewpoint of an application in the Application Layer, a change of the number of near-line material servers cannot be seen from the application in the Layer. As a result, any specific physical treatment of the near-line material servers such as increasing the number of it to accommodate an increase of materials can be conducted without influencing an application in the upper Application Layer at all.

Consequently, the UMID can be used to realize the loose coupling between the Application and Media Layers in a SOA-based media production system, resulting in further enhancing the flexibility in the system.

Frequently asked questions

Finally, issues and/or topics identified during the study but not discussed in the main body of the study report are listed as the frequently asked questions in the annex of the report, including the question on how UMID differs from other existing identification schemes for audiovisual entity such as ISAN or EIDR.

Study Report Part 2

Another report is under development in order to address the third project scope, or "To identify relevant technologies needed to be additionally standardized", which currently has the following three topics.

UMID Resolution Protocol

This is a common conversation method between an application and material servers demonstrated in Figure 4 in order to convert a given UMID into its corresponding URL. An intensive feasibility study of the DNS (Domain Name System) for the UMID resolution has been conducted, which is further discussed in the next section.

UMID based Program Package Exchange

The program package is a tool to bundle files relevant to a certain program such as MXF files, metadata files and thumbnail files into a single package. Because the program package has portability and can move through workflow or between organizations, it needs to be uniquely identified during its lifecycle regardless of its location. The UMID use for the program package identifier has been carefully studied based on its typical use case scenario.

UMID Applications for MXF

This is a study of UMID applications specifically for the MXF technology, While UMID is specified as a mandatory component at the MXF header metadata, its treatment in a practical situation has been unknown so far. By analyzing the existing MXF applications, this study will clarify how to treat UMID in an MXF file according the UMID Application Principles and explore parts that need to be additionally industry standardized.

RP 205 Revision AHG[6]

Based on the recommendation from the Study Report Part 1[5], a new activity has started to revise the existing SMPTE RP 205[2] to reflect the study result in the report including the UMID Application Principles. This is necessary because the study report itself is no more than just a private document effective only within SMPTE SC.

SMPTE RP 205 was originally provided to describe the recommended application of the UMID and to include requirements for the unique identification (tagging) of AV material to enable its reliable access and tracking at appropriate levels of granularity. Although it has successfully shown various interesting UMID applications and suggested some requirements on the UMID uses, it fails to clearly specify the most fundamental rules all UMID-aware products must strictly follow, that is, the UMID Application Principles. Furthermore, the distinction of the normative and informative parts of RP 205 is ambiguous.

Therefore, in the revised RP 205, the UMID Application Principles are clearly specified as normative part of this SMPTE standard document. Furthermore, useful information on the UMID applications obtained via the study, such as the UMID Managed Domain and the UMID application examples, are to be incorporated into the document as its annex so that they are shared among the industry.

At the time of this writing, the final working draft of the revised RP 205 has been submitted to the TC-30MR for the pre-ballot review, which is under careful review by the SMPTE experts. Hopefully, the revised RP 205 is expected to be published within early next year unless unexpected happens.

DNS-based UMID Resolution

Introduction

In this section, the latest study result of the DNS for the UMID resolution is briefly reported.

What is the DNS (Domain Name System)?

The DNS is a hierarchical, distributed, database system that constitutes the name resolution infrastructure of the Internet, which is primarily designed to associate network based resources with an easy to remember, human-readable, domain name, known as a hostname, and to translate those hostnames into the network resource's numerical network address or IP address.

For example, in order for the SMPTE web site (www.smpte.org) to be accessed by a web browser, the hostname "www.smpte.org" must be translated into the IP address that is associated to the "www" hostname within the "smpte.org" domain (107.21.249.46).

DNS is comprised of two sub-systems; the DNS Name Server, which manages the domain hierarchy and name resolution, and the Resolver, which acts as the client for the name server, issuing name resolution requests. It is the DNS Name Server that implements the distributed, hierarchical database structure. Figure 7 below schematically demonstrates how the domain name "www.smpte.org" is resolved to the IP address 107.21.249.46.

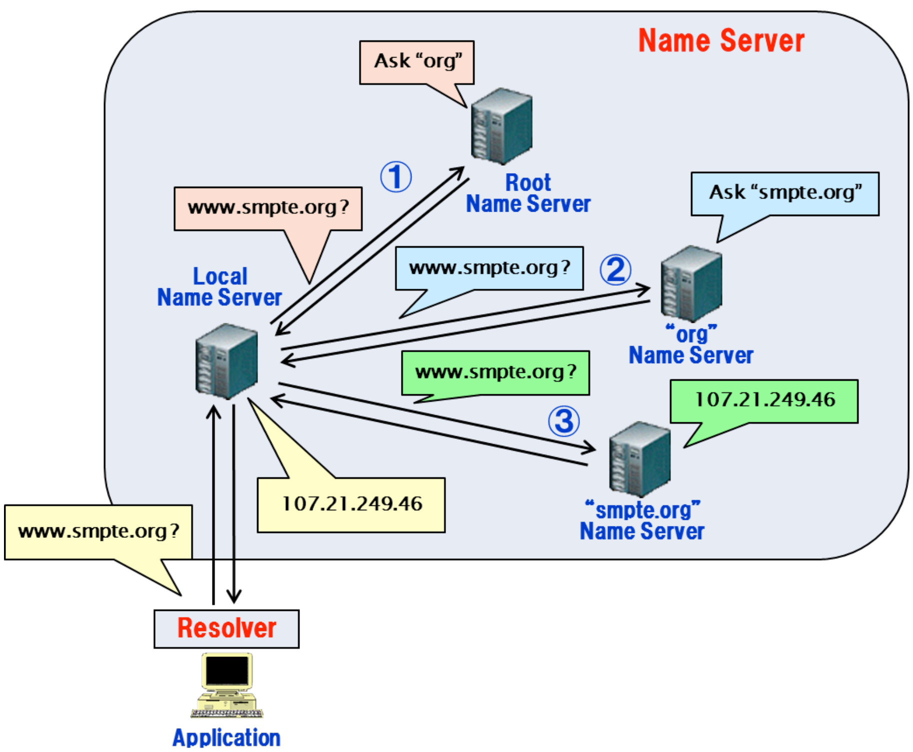


Figure 7: DNS Name Resolution for "www.smpte.org"

As shown in this figure, when an application ("Application") desires to obtain an IP address for the hostname "www.smpte.org", the application's Resolver issues the name resolution request to the pre-configured Local Name Server (DNS Name Server). If the Local Name Server already has a response for the request cached within its memory, it will immediately return the response.

If the Local Name Server does not have the response cashed, however, it will delegate the name resolution request to its pre-configured Parent Name Server. If the Parent Name Server does not have the response cashed, it will further delegate the name resolution request to its pre-configured Parent Name Server recursively until either a response is made known or the Root Name Servers, defined for the Internet, are reached.

The Root Name Servers are DNS Name Server assigned by the Internet Assigned Numbers Authority (IANA) to manage the Top Level Domains (TLDs), such as ".com", ".org", and ".net". Since the TLD for the SMPTE web site's hostname ("www.smpte.org") is ".org", the Root Name Server will delegate the request to the Name Server assigned as the Start of Authority (SOA) for the ".org" TLD, which will then further delegate the request to the Name Server assigned as the SOA for the "smpte.org" domain.

Because "www" is a hostname provided within the "smpte.org" domain, the Name Server assigned as the SOA for the domain also authoritatively manages the correspondence between the hostname "www" and its IP address 107.21.249.46, which will be finally returned as a response to the Resolver via the Local Name Server that will cache the response in preparation for the future request.

While this recursive behavior of DNS is a critical feature for the scalable and extensible of the name resolution system, each DNS Name Server can be described as a simple query/response database from the vantage point of the Resolver, *i.e.*, the Resolver issues a name query request to the Local Name Server and the Local Name Server replies to the request, after performing the recursive process described above.

DNS-based Service Discovery (DNS-SD)[12]

While the DNS is usually used for the name resolution, it can also be used as a general, distributed, hierarchical database, storing almost any kind of data. Thanks to the DNS as a well-known and widely accepted technology, various trials have been made to utilize the DNS for other purposes than the name resolution, among which one of the most widely accepted applications is the DNS-SD (DNS-based Service Discovery).

The DNS-SD is a means of using the DNS to search the network for services. The DNS-SD allows a client to search for a desired service type and the client will receive a list of specific service instances. The client can then choose from the service instances in an automated fashion or through user interaction.

Figure 8 schematically demonstrates how an application in a local office network finds a particular service instance such as a certain web page based on the DNS-SD.

In Figure 8, an application ("Application") desiring to access "My web page" distributes a query asking "Where is 'My web page'?" by using the multicast DNS[13], another kind of DNS that uses the multicast instead of the unicast used for the conventional DNS, and the web server connected to the local office network ("Web Server") will respond to it with the URL as an entry point to the "My web page", which is then to be used for the application to actually access the web page.

It should be noted that each device connected to the network can directly talk with the application for this example, indicating that it contains the DNS Name Server that manages service instances and their access information provided on its own. Therefore, it is also possible for the application to sequentially ask each device one by one for a desired service instance by using the conventional unicast DNS.

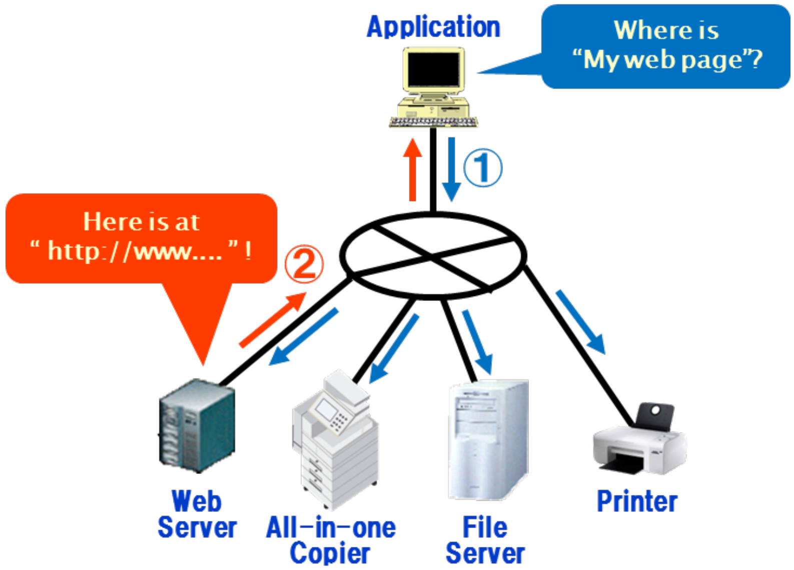


Figure 8: DNS-SD in a Local Office Network

DNS-based UMID resolution

Basic Concept

Compared the conversation between the application ("Application") and the web server ("Web Server") in Figure 8 with that between the application ("Application") and the ingest server ("Ingest Server") in Figure 4 (UMID based material search), one can easily notice that they are quite similar. This is the basic concept of the DNS-based UMID resolution in a local network.

Feasibility Study of DNS for UMID Resolution

Suppose a material uniquely identified by the UMID of 06_h0a_h … $0d_h20_h$ … bf_hdf_h is stored in the ingest server in a local network (`ingSrv.local`) as an MXF file (`file5.mxf`) at a designated directory (`/20130831/sports/`). Suppose also as for its basic technical properties, an essence of the material is encoded by JPEG 2000 (`codec=j2k`), wrapped to form an MXF file (`wrapper=mxf`), whose playout exhibits 60 second duration (`dur=60`) with the 10 bits/pixel (`depth=10`) and the YUV 422 color subsampling (`samle=422`). Then, the correspondence between the UMID and its basic technical properties including the corresponding URL can be described by using the DNS Text Record as:

```
0d20.13000000.d2c9036c.8f195343.ab7014d2.d718bfdf._umid._udp.local
        IN TXT ("dur=60" "depth=10" "sample=422")
0d20.13000000.d2c9036c.8f195343.ab7014d2.d718bfdf._umid._udp.local
        IN TXT ("url=http://ingSrv.local/20130831/sports/file5.mxf"
                   "codec=j2k" "wrapper=mxf")
```

When "BIND", the well-known implementation of the DNS Name Server, is configured with the above Text Record descriptions, it can work as the UMID Name Server that resolves a given UMID into its corresponding URL together with its basic technical properties. Figure 9 demonstrates the DNS-based UMID resolution by using the "BIND" running in the background as the UMID Name Server after the aforementioned configuration at its execution and "nslookup", the widely available implementation of the DNS Resolver, as a client to ask the UMID Name Server for the UMID resolution.

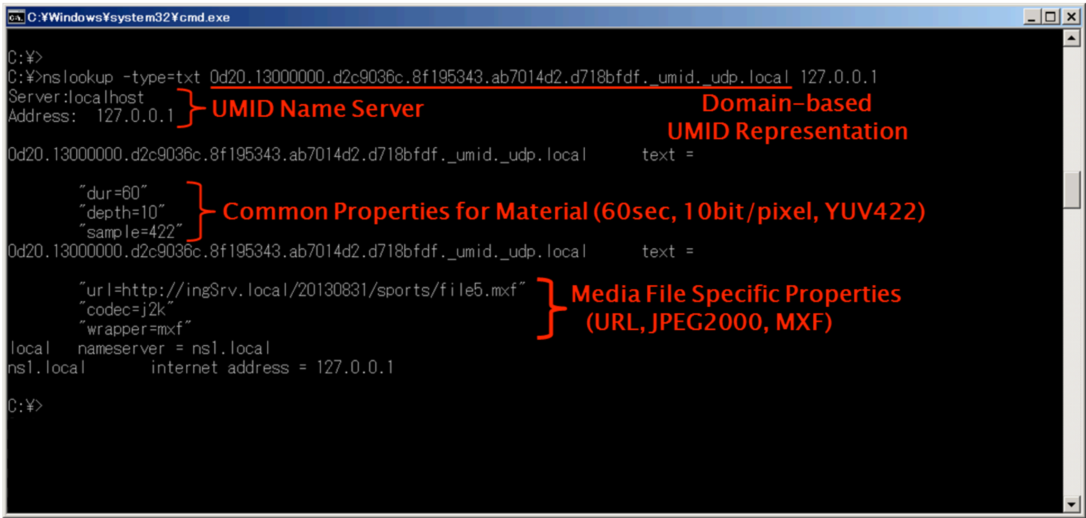


Figure 9: DNS-based UMID Resolution using "BIND" and "nslookup"

In this figure, "nslookup" is executed with the argument,

```
0d20.13000000.d2c9036c.8f195343.ab7014d2.d718bfdf._umid._udp.local
```

which is in fact a domain-based representation of the UMID following the DNS-SD convention, and a type option specifying a query for the DNS Text Record type. The response received from the UMID Name Server (`localhost`) shows two distinct groups of the `key=value` text strings: the one containing duration (`dur`), pixel bit depth (`depth`) and color subsampling (`sample`) as the keys, and the other containing codec (`codec`), wrapper (`wrapper`) and the URL (`url`) composed of the `ingSrv.local` ingest server and the file path.

Note that it is important for the basic technical properties that are independent of a specific instantiation of a material as a media file, such as duration, pixel bit depth and color subsampling, to be grouped separately from those specific to the media file (codec, wrapper and URL in this example) in order to describe the intrinsic properties of the material at its playout, which is actually what the UMID uniquely identifies, separately from those specific to its instantiation as a media file. Consequently, even if a clone of the material is additionally created and stored in the near-line material server (`matSrv.local`) as an MXF file (`mat5.mxf`) at a designated directory (`/program1/materials/`), it is reflected to the UMID Name Server by additionally configuring it with the DNS Text Record description as:

```
0d20.13000000.d2c9036c.8f195343.ab7014d2.d718bfdf._umid._udp.local
          IN TXT ("url=http://marSrv.local/program1/materials/mat5.mxf"
                  "codec=j2k" "wrapper=mxf")
```

which will give additional group of the `key=value` text strings for those technical properties in the response from the UMID Name Server.

Why is the DNS-based used?

The DNS is the most widely accepted name resolution system utilized in IP networks and is typically present on all IP networks. DNS is also well understood by IT professionals and DNS name resolution clients are built into most modern operating systems.

While the benefits of use of DNS for the service discovery are intensively discussed in the literature[12], they are in fact applicable to the DNS-based UMID resolution basically as they are, *i.e.*, when they are rewritten specifically for the UMID resolution, they would become as follows:

- UMID resolution requires an aggregation server that manages the correspondence between a UMID and a URL of a material identified by the UMID (the UMID and URL correspondence). DNS already has the one: a DNS Name Server (UMID Name Server).
- UMID resolution requires the UMID and URL correspondence registration protocol. DNS already has the one: DNS dynamic update[14].
- UMID resolution requires a query protocol. DNS already has the one: DNS queries.
- UMID resolution requires security mechanisms. DNS already has security mechanisms such as DNSSEC[15].
- UMID resolution requires a multicast mode for a system of a closed local network as shown in Figure 4. DNS already has the one: the multicast DNS (mDNS)[13].

In summary, the DNS-based UMID resolution can make use of already existing servers, clients, protocols, infrastructure, tools, and expertise for the DNS, which is mature technology, well understood, with multiple independent implementations from different vendors, a wide selection of books published on the subject, and an established workforce experienced in its operation.

Furthermore, the advanced and/or future requirements for the UMID resolution such as fault-tolerance, backward compatibility, extensibility, reliability, maintainability, availability, security, usability, and so on, have been already satisfied with, or will be addressed by if insufficient, the extensions of the DNS technology.

It should be noted that, in order to obtain full advantage of the use of DNS, the DNS-based UMID resolution should be designed so that no changes are made to the DNS message structure. The DNS-based UMID resolution should also use existing DNS Resource Record (RR) types as they are so that there is no need to define new RR types or operation codes.

Conclusions

In this paper, the latest status of the UMID Application Project in SMPTE is reported. The UMID Application Principles, the fundamental rules all UMID-aware products must strictly follow, have been already identified and SMPTE RP 205 is under revision in order to normatively specify the Principles. The revised RP 205 will also include useful information worthwhile to be shared in the industry such as the UMID Managed Domain as an embodiment of the UMID Application Principles and various UMID application examples collected in the project.

It is also reported that the feasibility study of the DNS for the UMID Resolution, a method to convert a given UMID into its corresponding URL, has revealed that the UMID resolution will be successfully realized by utilizing the DNS technology, while basic technical properties that should be included into the response need to be additionally specified as the SMPTE standards.

In 1998, the EBU/SMPTE Task Force recommended to industry standardize a unique material identifier and a single generic file wrapper for the streaming of essence and metadata for the exchange of program material as bit-streams[16]. They have been already realized as UMID[1] in 2000 and as MXF[7] in 2004, respectively. It is therefore a natural consequence for the UMID to be introduced into the MXF file as a mandatory component to uniquely identify it.

As the file-based workflow becomes common, the MXF files have been widely accepted so far in the industry. This suggests that UMID is also already widely spread by using an MXF file as a vehicle, though it is useless at all for now. But such a situation will be drastically changed when additional technologies to enhance the UMID applications are to be industry standardized by SMPTE, which should come shortly.

Finally, those interested in the UMID Application Project are kindly requested to join us and to contribute their expertise to this future-proof activity.

Acknowledgements

The author would like to thank Mr. Jim Wilkinson (Wellspring Digital) for his valuable assistance in the preparation of this paper. He also appreciates all members in the UMID Application Project for their valuable contributions they have made so far and their continued support.

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 Y. Shibata and J. Wilkinson, “UMID Applications in Practices”, Proc. SMPTE 2011 Annual Technical Conference (in “Topics In File Based Work Flows (Part 3)”) (2011)

4 TC-30MR SG UMID Applications https://kws.smpte.org/kws/public/projects/project/details?project_id=90

5 STUDY REPORT ON UMID APPLICATIONS PART 1 - UMID Application Principles and Best Practices, (SMPTE SC members only) available at: https://kws.smpte.org/kws/groups/30mr/download.php/22276/Study%20Report%20on%20UMID%20Applications%20Part%201%20Rev1.pdf

6 RP 205 Revision: UMID Application https://kws.smpte.org/kws/public/projects/project/details?project_id=174

7 SMPTE Standard ST 377-1:2011 Material Exchange Format (MXF) — File Format Specification

8 S. Takeuchi, et al., “A Program Production System Using ID and File-Data Over IP Networks”, SMPTE Mot. Imag. J.; April 2005; 114:(4) 132-138

9 M. De Geyter, et al., “Integration Demands on MAM Systems: A Proof of Concept Solution,” SMPTE Mot. Imag. J.; Nov./Dec. 2008; 117:(8) 38-46.

10 Y. Shibata, et al., “Introduction to XDCAM Metadata”, Proc. NAB 2005 Broadcast Engineering Conference, pp.421-424 (2005)

11 The Framework for Interoperable Media Services (FIMS): http://fims.tv/

12 Internet Engineering Task Force (IETF) (2013, February). IETF RFC 6763 – DNS-Based Service Discovery (DNS-SD): http://tools.ietf.org/html/rfc6763.html

13 Internet Engineering Task Force (IETF) (2013, February). IETF RFC 6762 – Multicast DNS (mDNS): http://tools.ietf.org/html/rfc6762.html

14 Internet Engineering Task Force (IETF) (1997, April). IETF RFC 2136 – Dynamic Updates in the Domain Name System (DNS UPDATE): http://tools.ietf.org/html/rfc2136.html

15 Internet Engineering Task Force (IETF) (2005, March). IETF RFC 4033 – DNS Security Introduction and Requirements: http://tools.ietf.org/html/rfc4033.html

16 EBU Technical Review Special Supplement 1998, “EBU / SMPTE Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams, Final Report: Analyses and Results”: http://tech.ebu.ch/docs/techreview/ebu-smpte-tf-bitstreams.pdf

Latest Status of UMID Application Project in SMPTE

Yoshi Shibata, Chair, SMPTE TC-30MR SG UMID Applications
(metaFrontier.jp, LLC, Yokohama, Japan)

Outline

- Introduction
- What is the UMID?
- UMID based Material Search
  - UMID Resolution Protocol
  - UMID Application Principles
- UMID Application Project
  - Study Report Part 1
  - Study Report Part 2
- DNS-based UMID Resolution
- Conclusions

Introduction

- Happened since two years ago …

“UMID Applications in Practice”

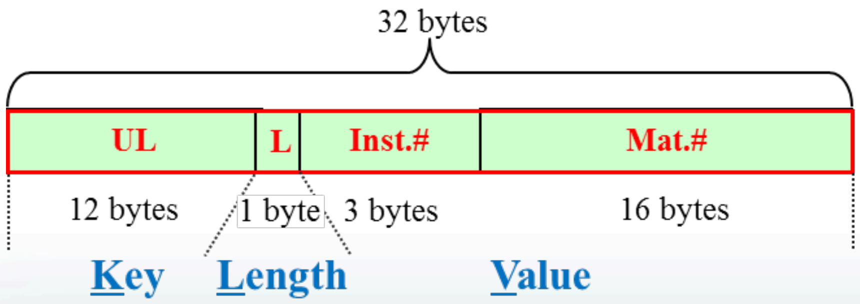
By Yoshi Shibata & Jim Wilkinson

The 2011 Annual Technical Conference & Exhibition

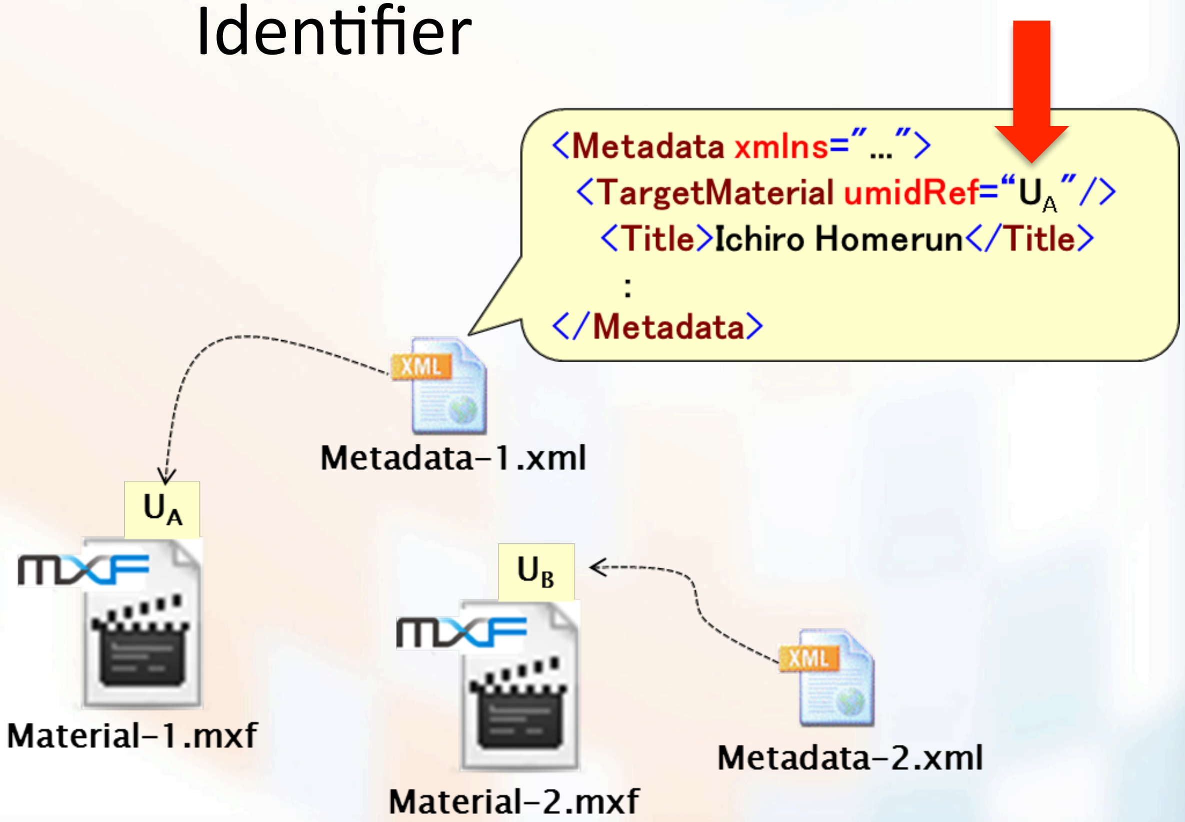
SMPTE Home Page (Nov. 25, 2011)

What is the UMID?

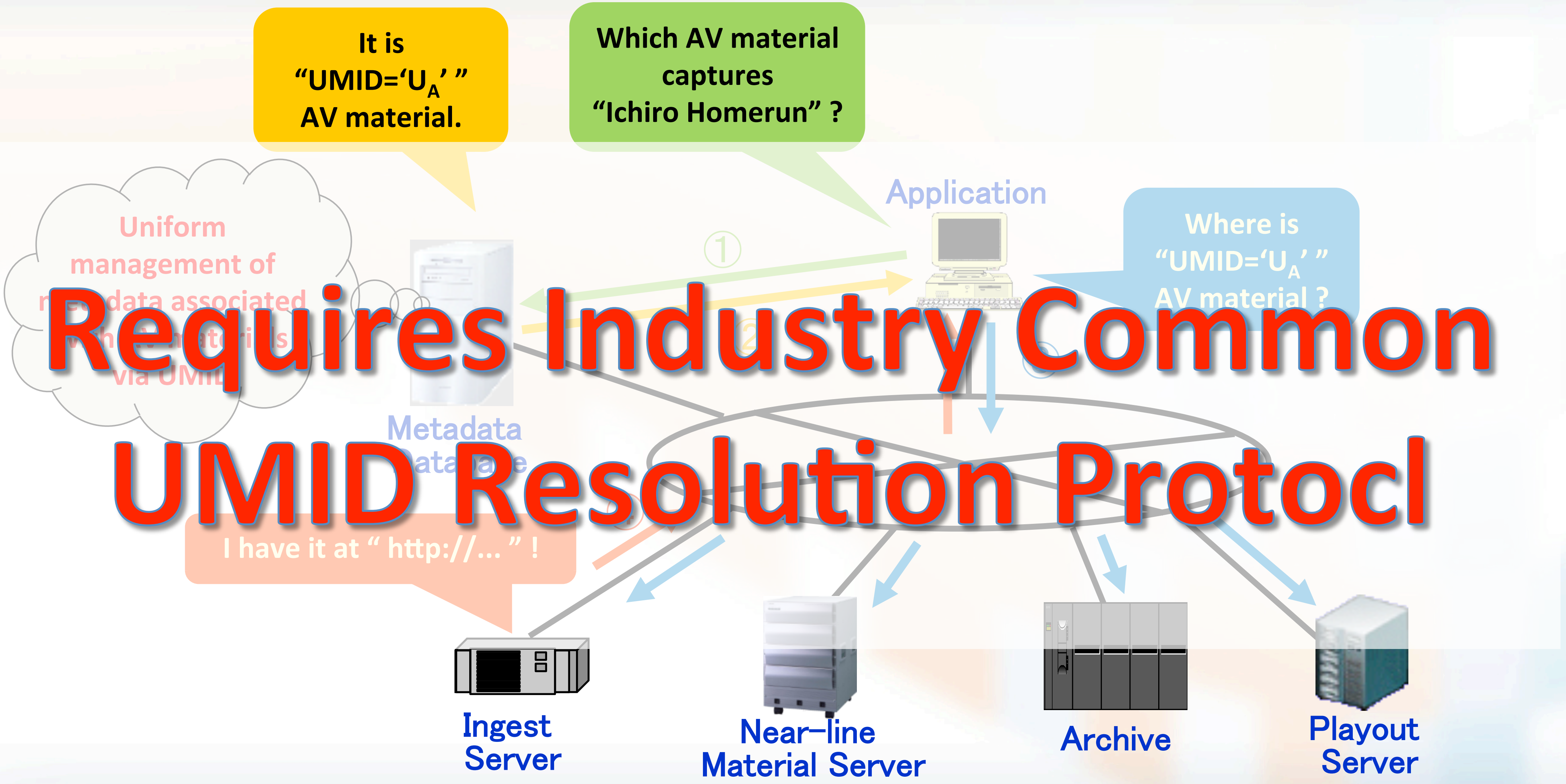
- Unique Material IDentifier
  - For Audiovisual (AV) Material
- UMID Characteristics
  - SMPTE Standard (ST 330)
  - Mandatory item for an MXF file
- UMID Structure



- Typical UMID Use
  - As a Globally Unique Material Identifier



UMID based Material Search



Requires Industry Common UMID Resolution Protocl

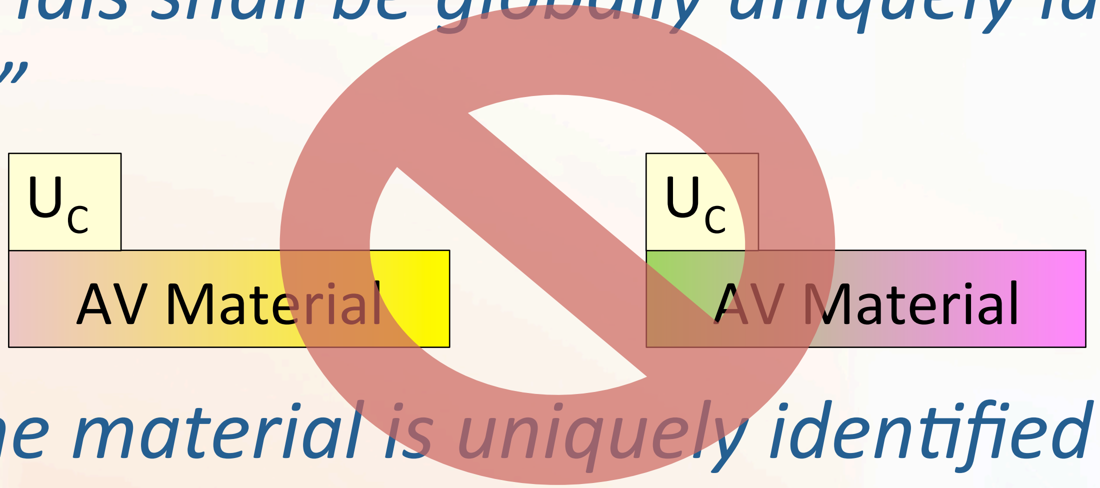
How to Realize It?

- Define Industry Common **UMID Resolution Protocol**
  – To convert a given UMID of a desired AV material to its corresponding URL

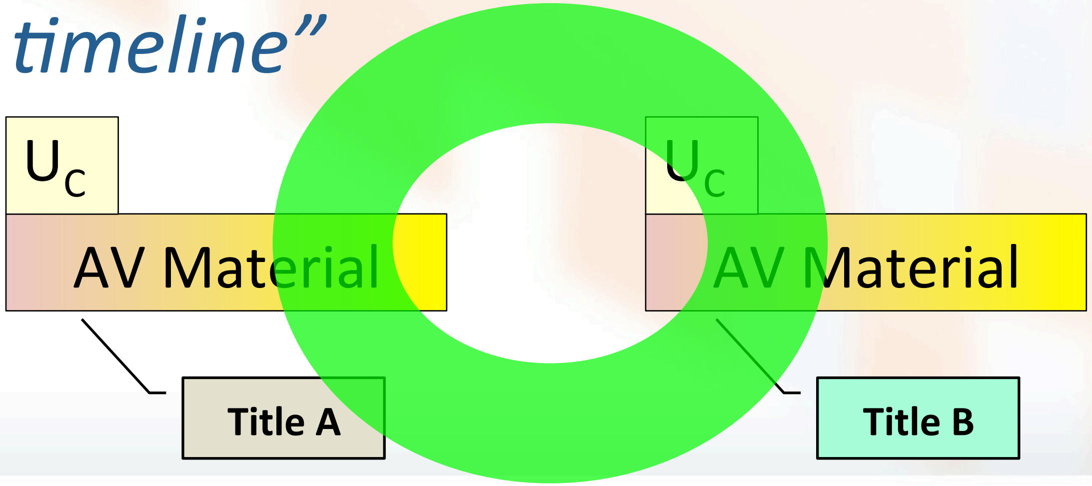
- Clearly Specify **UMID Application Principles**
  – To realize UMID based AV material management

What are the UMID Application Principles?

- *"Different Materials shall be globally uniquely identified by different UMIDs"*



- *"If more than one material is uniquely identified by a single UMID, their representations at playout shall be identical bit-by-bit on the timeline"*



UMID in Media Production System

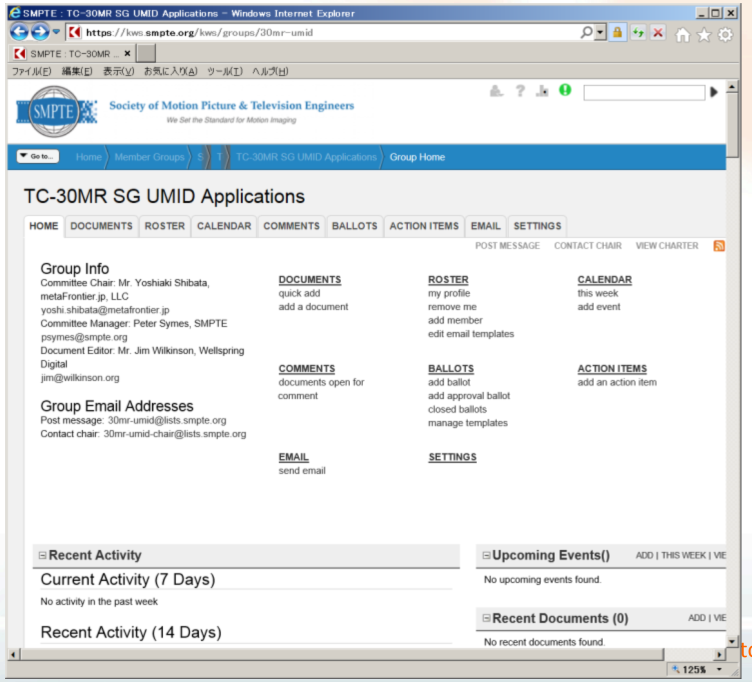
UMID as a Common AV Material Identifier In "Best-of-Breed" System

UMID Ready

UMID Application Project

TC-30MR SG UMID Applications

- Chair
  - Yoshi Shibata
- Main Editor
  - Jim Wilkinson
- # of Participants
  - 47 (as of Oct. 10, 2013)



- History
  - 2012
    - Proposed (Mar. 8)
    - Approved (Apr. 10)
    - Kick-off (Jun. 11)
    - UMID Application Conference (Sep. 13)
  - 2013
    - Submitted Study Report Part 1 (Feb. 4)
      - UMID Application Principles
    - Established RP 205 Revision AHG (May 5)
    - Developing Study Report Part 2 (now)
      - DNS-based UMID Resolution

Study Report Part 1 (1/4)

Contents:

- Proposal on
  - UMID Application Principles
- Introduction to
  - UMID Managed Domain
  - UMID Application Examples
- Frequently Asked Questions

Study Report Part 1 (2/4)

- UMID Application Principles
  - Principle 1: Definition
  - Principle 2: UMID Creation
  - Principle 3: UMID Integrity
  - Principle 4: UMID Identification
  - Principle 5: UMID Inheritance
  - Principle 6: Extended UMID
  - Principle 7: Source Pack

Study Report Part 1 (2/4)

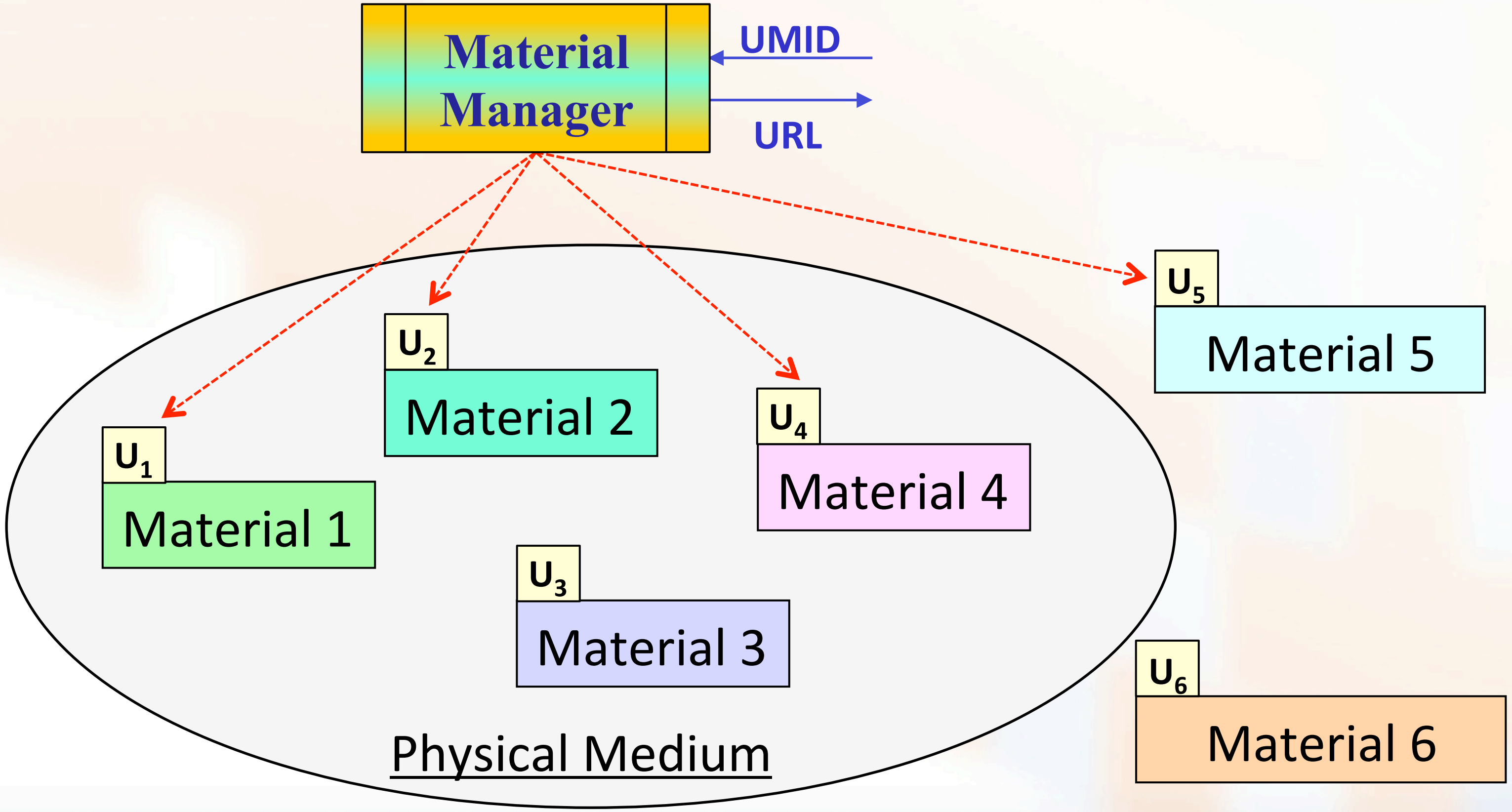
- UMID Application Principles
  - Principle 1: Definition
  - Principle 2: UMID Creation
  - Principle 3: UMID Integrity
    - ***Different Materials shall be globally uniquely identified by different UMIDs.***
  - Principle 4: UMID Identification
  - Principle 5: UMID Inheritance
  - Principle 6: Extended UMID
  - Principle 7: Source Pack

Study Report Part 1 (2/4)

- UMID Application Principles
  - Principle 1: Definition
  - Principle 2: UMID Creation
  - Principle 3: UMID Integrity
    - ***Different Materials shall be globally uniquely identified by different UMIDs.***
  - Principle 4: UMID Identification
    - ***If more than one material is uniquely identified by a single UMID, their representations at playout shall be identical bit by bit on the timeline.***
  - Principle 5: UMID Inheritance
  - Principle 6: Extended UMID
  - Principle 7: Source Pack

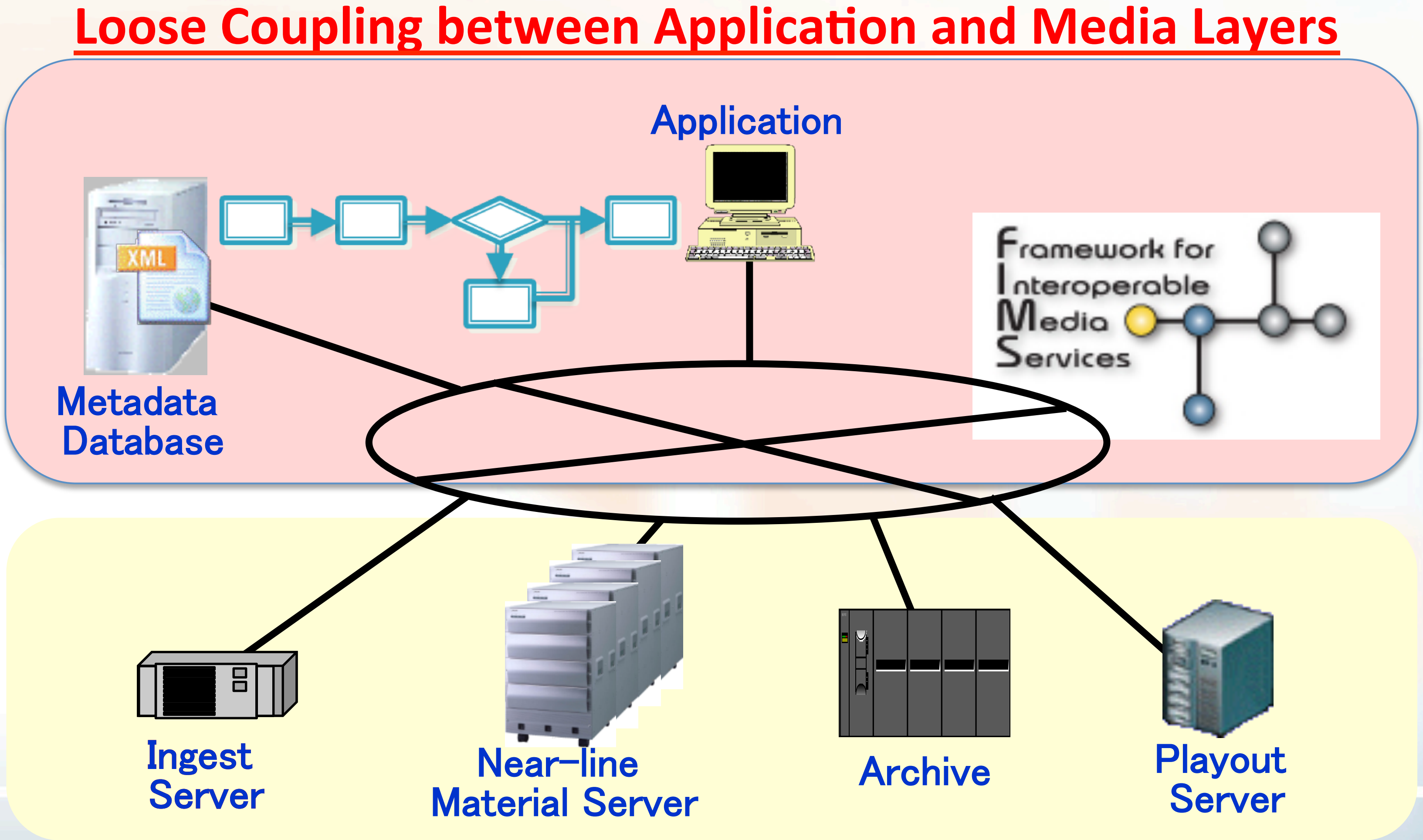
Study Report Part 1 (3/4)

- UMID Managed Domain
  - As an embodiment of "UMID Application Principles"



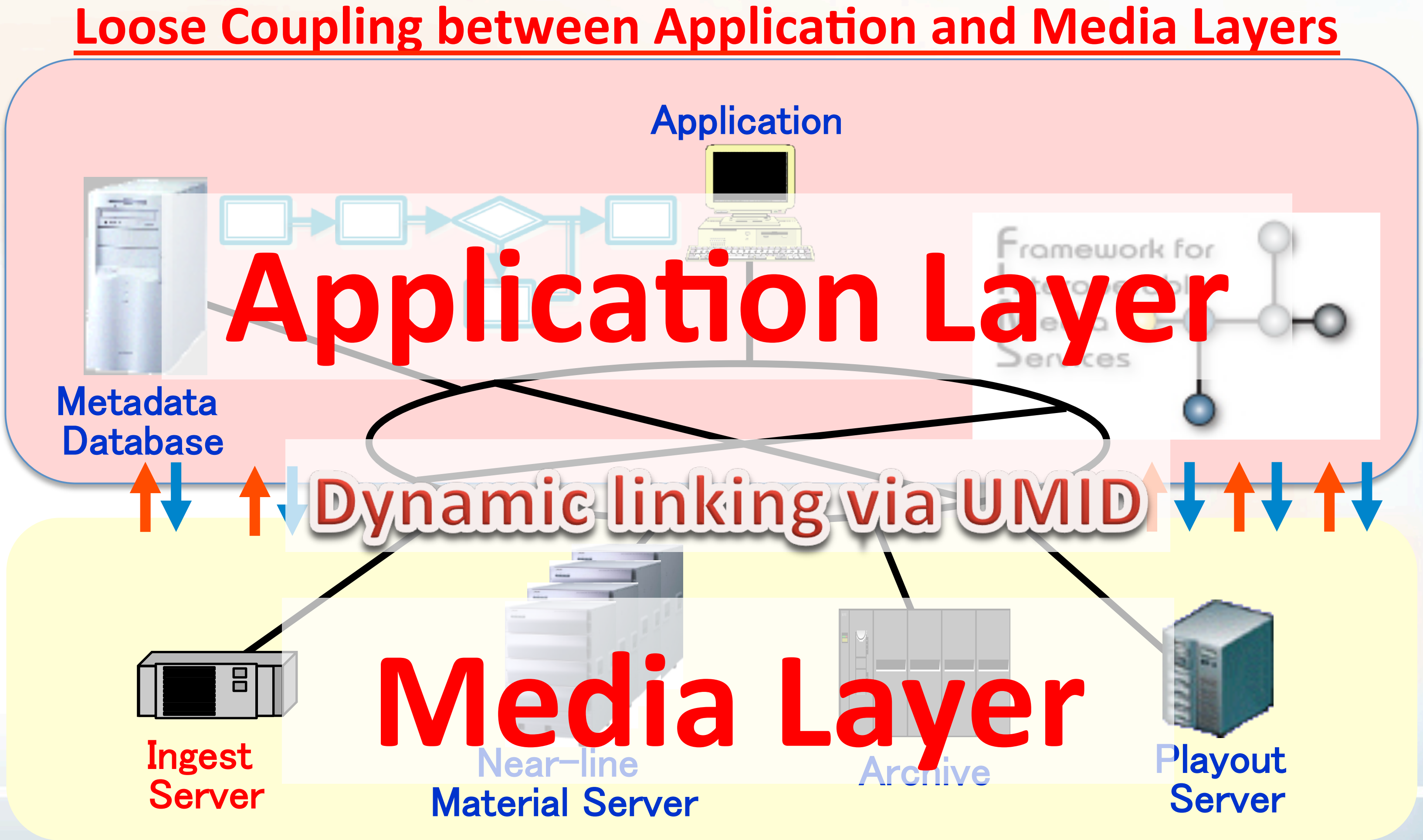
Study Report Part 1 (4/4)

- UMID Application Examples
  - More than a dozen examples introduced



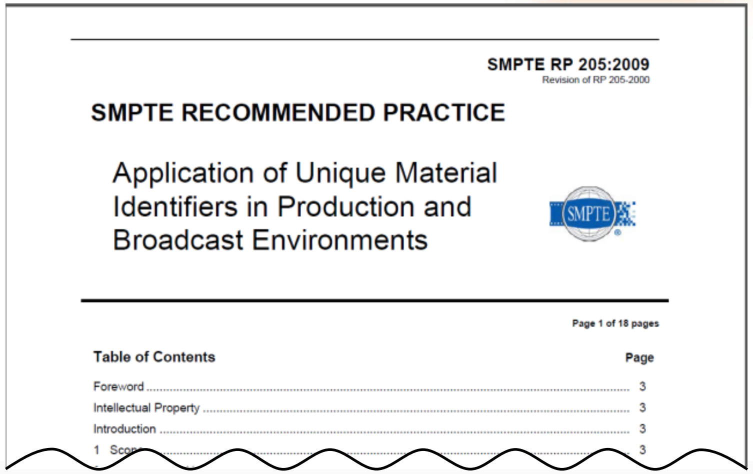
Study Report Part 1 (4/4)

- UMID Application Examples
  - More than a dozen examples introduced



SMPTE RP 205 Revision AHG

- Setup on May. 16, 2012
- To revise RP 205:2009
  - based on the recommendation from Study Report Part 1
- Chaired by Yoshi Shibata
- Under pre-FCD-ballot review now



SMPTE RP 205:2009
Revision of RP 205-2000

SMPTE RECOMMENDED PRACTICE

Application of Unique Material Identifiers in Production and Broadcast Environments

Page 1 of 18 pages

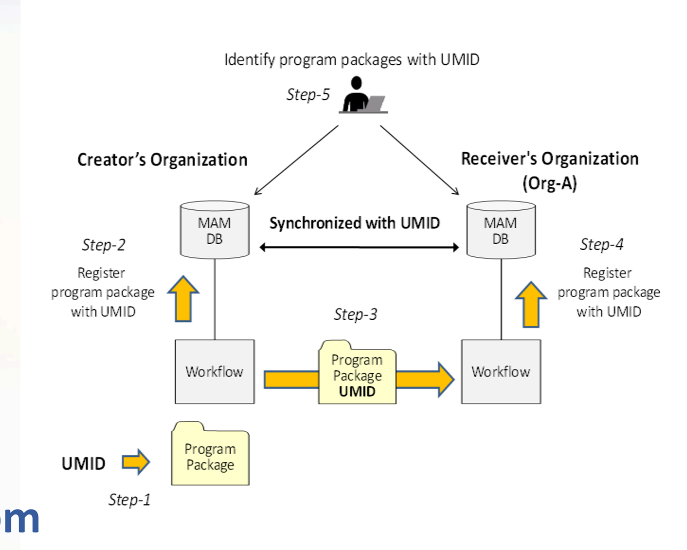
Table of Contents

SMPTE RP 205:2009

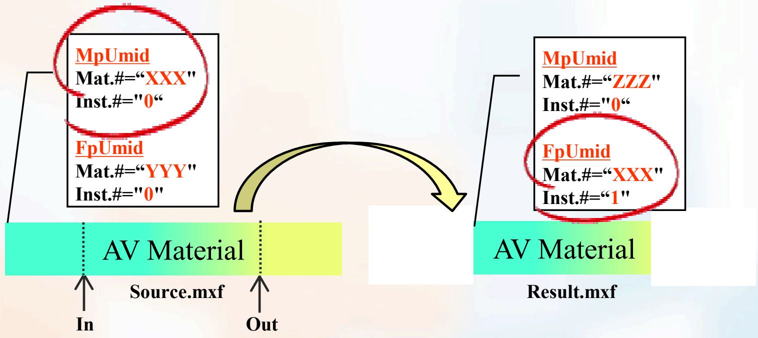
- Contents of proposed RP 205
  - UMID Basics (Informative)
  - UMID Application Principles
  - Considerations on the UMID Uses in its Applications
  - Annex (Informative)
    - How to implement the UMID Application Principles
      - UMID Managed Domain
    - UMID Application Examples
    - Frequently Asked Questions

Study Report Part 2

- Under development
  - Additional Technologies needed to be industry standardized
- Topics to be included
  - UMID Resolution Protocol
    - To be discussed later
  - UMID based Program Package Exchange
    - To bundle relevant files for delivery
  - UMID Applications in MXF
    - To tailor UMID Application Principles for MXF



From
"UMID based Program Package Exchange"

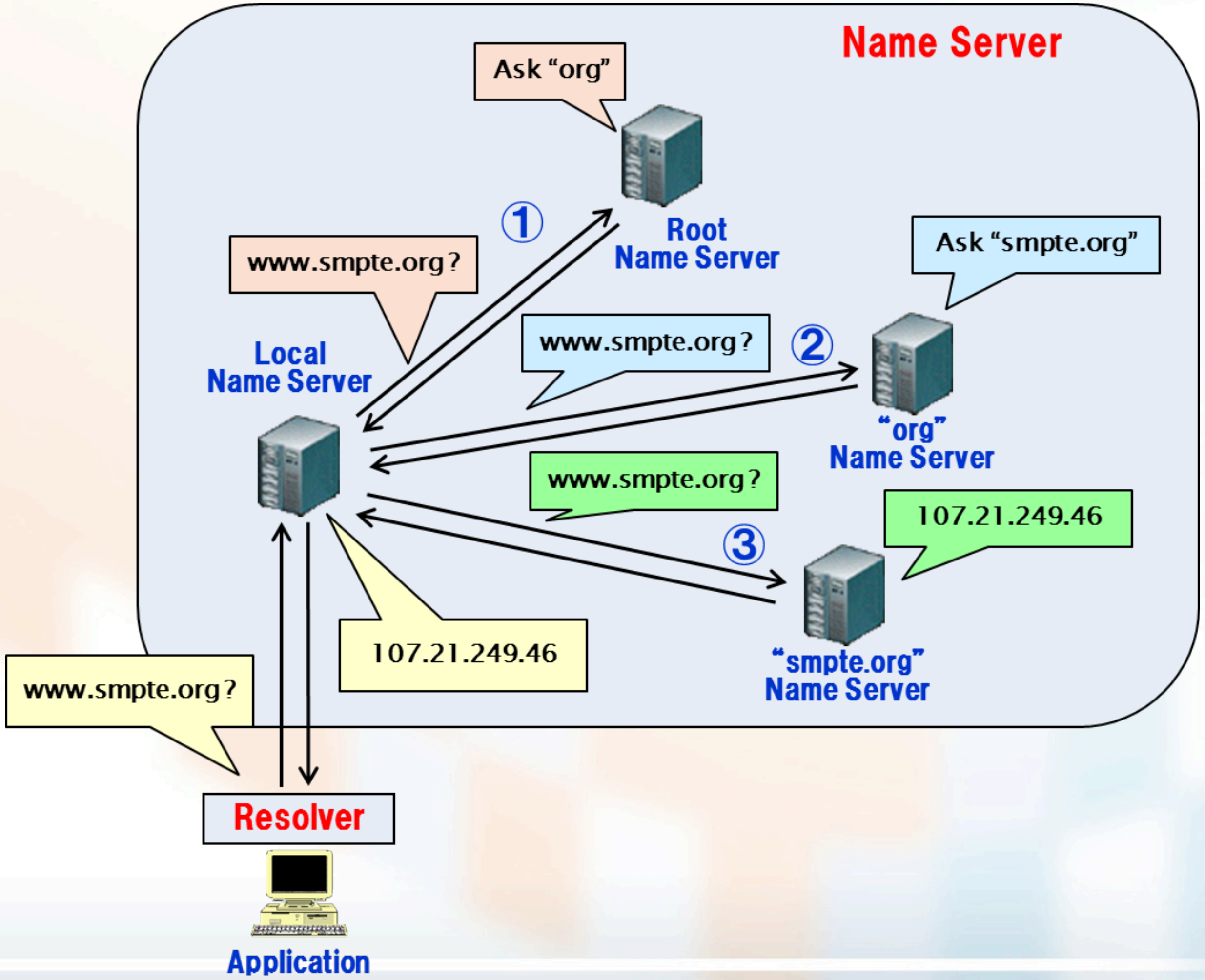


From
"UMID Applications for MXF"

DNS-based UMID Resolution

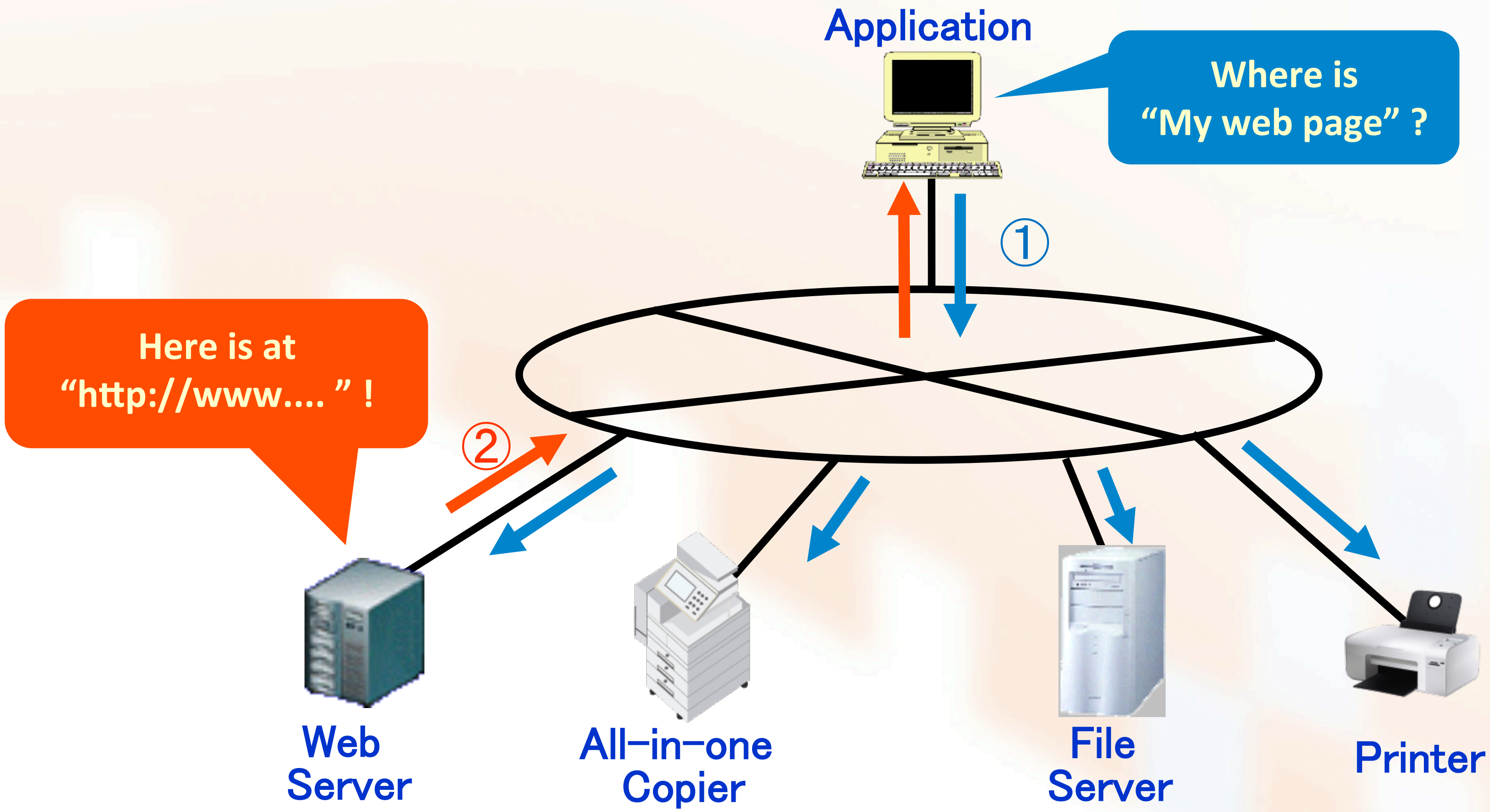
What is the DNS?

- Domain Name System
  - As the name resolution infrastructure for the Internet
    E.g.,
    “www.smpte.org”
    ↓
    “107.21.249.46”
  - Hierarchically distributed database system
  - Can be used as a general database for any purpose!
- Name Resolution



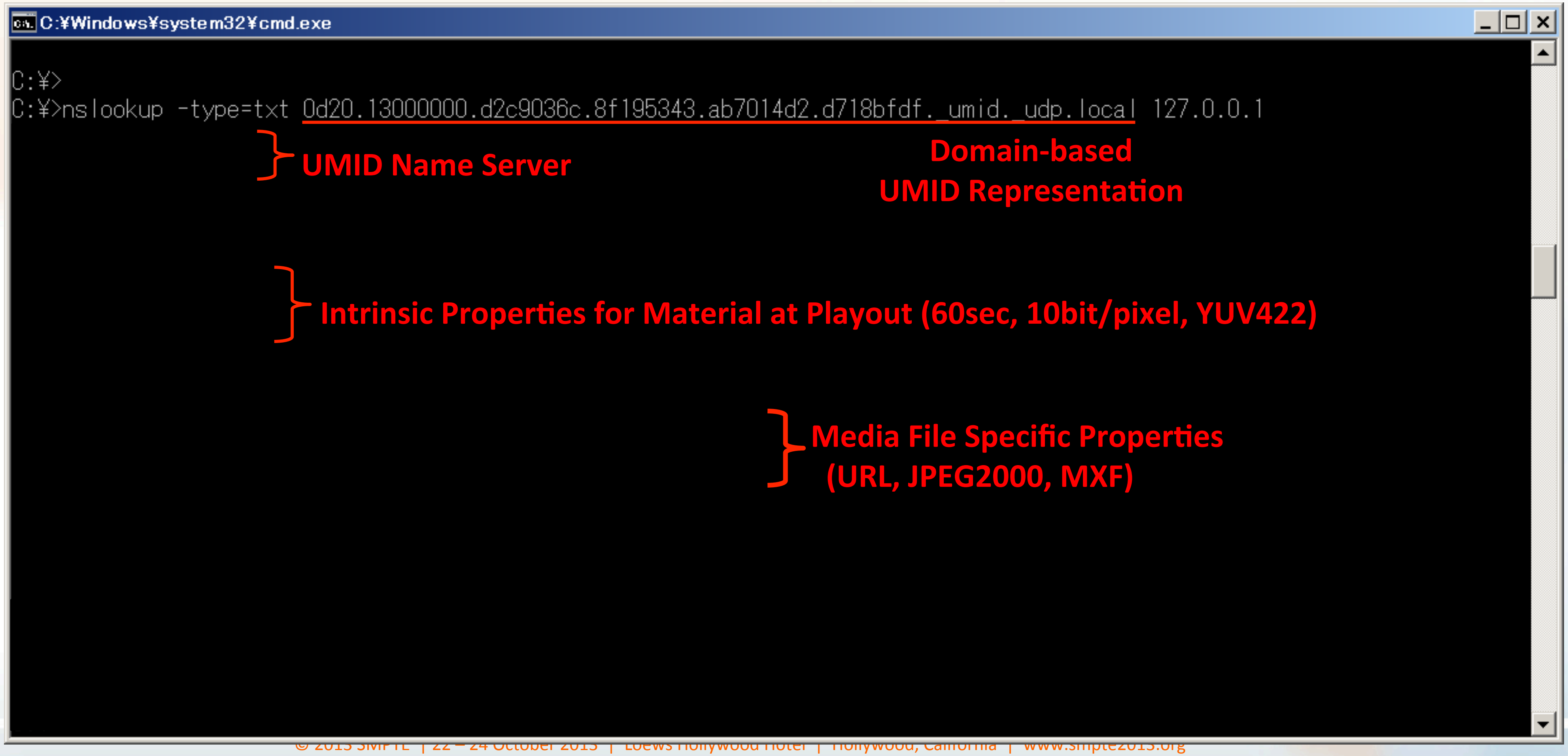
DNS-based Service Discovery (DNS-SD)

- For Application to Discover a Particular Service Instance



DNS-based UMID Resolution

- Preliminary Experimental Result
  - Simulation using “BIND” (Server) and “nslookup” (Client)



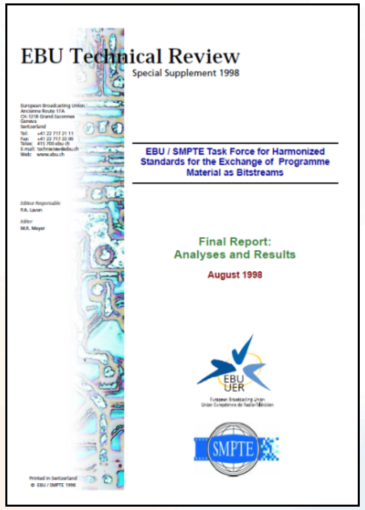
Why is the DNS-based used?

- The most widely accepted name resolution system
  - A lot of implementations, books, seminars and experts
- Can make use of already existing technology
  - Name Server, Resolver (Client), Protocols (query/response/update), tools (tester, analyzer, etc.), and so on
- Will enjoy already existing or future extensions
  - fault-tolerance, backward compatibility, extensibility, reliability, maintainability, availability, security, usability, and so on

As long as the DNS-based UMID resolution uses DNS message structure, DNS Resource Records and DNS operation code <u>as they are</u>.

Conclusions

- Reported the latest status of UMID Application Project
  - **UMID Application Principles** to be standardized in new SMPTE RP 205
  - **UMID Resolution Protocol** to be realized based on DNS
- Btw, Why UMID???
  - Originally proposed and standardized by SMPTE
    - Together with MXF ➔ SMPTE 330M-2000
  - Mandatory Item for an MXF file
    - Your MXF files already have UMID! (Useless so far, though)
  - **We will change the situation shortly!**



EBU/SMPTE Joint Task Force Final Report (1998)

PLEASE JOIN US!

(info@metafrontier.jp)