A METHOD FOR IMPLEMENTING UNIQUE IDENTIFIERS IN THE ENTERTAINMENT SUPPLY CHAIN

Michael Papish, Solutions Architecture Director, Rovi Corporation
Adam Powers, VP, Technology Strategy, Rovi Corporation

Abstract:

The emergence of digital technologies has transformed every aspect of the professional audiovisual supply chain from content creation and post-production to distribution and consumption and created new opportunities for all stakeholders. With these opportunities also come challenges, such as more complex value chain interactions and an explosion in the number of assets relevant to commerce.

In an effort to make the most of the new opportunities and address the challenges, major content producers have redesigned workflows to make them content-centric, putting in place digital media management infrastructure to help streamline operations. While these efforts help reduce inefficiencies, the role of unique identifiers in the movie and television supply chain can also facilitate and improve the efficiency of automated workflows.

In late 2010, the Entertainment Identifier Registry (EIDR) was launched by a group of companies — including Comcast, Disney, Warner Bros, Rovi, MovieLabs, and CableLabs — as a non-profit trade organization acting as a centralized registry for uniquely identifying video works. EIDR is built on the flexible data model established by the Digital Object Identifier standard.

However, with digitization has also come a fragmentation of the entertainment supply chain. As the industry adopts cloud-based systems, a web services model built on open APIs will soon follow. Small app-developers will vie with large hardware manufacturers to deliver the most engaging consumer entertainment experience using the same platforms and content items. We need to translate our centralized ID space into a technology service layer which can be accessed by a distributed ecosystem.

This paper provides an overview of the implementation of and interfaces to EIDR, including registering and looking up records, metadata schema, and deduplication process. We then extend this registry into the distributed ecosystem by describing technical methods developers can use to keep in sync with the ID space as it grows and morphs. Web service-based mechanisms for matching and translating content IDs across catalogs are described. We introduce the concept of ID stability and a time to live (TTL) model that uses readily understood concepts of internet architecture to implement EIDR and EIDR-like ID spaces into common software frameworks. We conclude with an investigation of potential implications and implementations for content management, advertising, interactive applications, and EPG metadata.

SECTION 1 – WHY UNIQUE IDS?

The past decade has seen the beginning of a dramatic shift in how content is created, distributed and consumed. Theatrical production costs have dropped while production workflows have become increasingly more complex and fragmented; broadcast workflows are making the switch to be file based; MSOs are reaching out to every screen with broadcast, video on demand, and new forms of engaging consumers; and advertisers are under pressure to deliver lower CPMs, higher targeting and more engagement.

This total upheaval of the entire supply chain has created many challenges, but also presents many opportunities as well. Whether a company's dream is to provide the myriad of new features and functionalities that are creating buzz on the market, or even achieving the long desired dream of total workflow automation, the devil is frequently in the details when reaching for those dreams. One of those devils is reaching a scale that allows those dreams to be economically viable, and the human factor is usually the largest hindrance. Even after several decades of personal computers and more than a decade of widespread Internet usage, humans are all too often involved in every step of processing and distributing content – manually creating avail notices, manually ingesting content, manually associating interactive assets with linear content.

In order to transition away from the manual infrastructure that is in place today to the automated infrastructure that enables tomorrow's applications and scale, new processes and techniques will need to be created that allow assets to be understood and processed by computers. Perhaps the largest hindrance is enabling computers to understand the assets with which they are dealing. In the manual world, humans have a great deal of context and outside information when working with assets. ―Which version of Robin Hood is this?," can often be answered with some understanding of whether a human is working with new releases, animations, or back catalog – although mistakes still happen. Computers, lacking context and nuanced decision making capabilities, must rely on unique identifiers for assets.

But all unique identifier systems are not created the same, nor can they all be applied the same. This paper attempts to lay out a framework for both designing and analyzing unique identifier systems. It then applies that analysis to two different systems: the unique identifier systems that have broad adoption in the market today, and the Entertainment Identifier Registry (EIDR) that was launched late last year.

SECTION 2 – THE DIFFICULTIES IN FORMING IDS

Whereas previously we could aspire to create a single, perfectly structured and completely clean database of entertainment content, we must now admit that this task is Sisyphean. Not only has the pace of content creation increased but the rise of semi-professional forms of audiovisual content such as YouTube complicates the matter. In addition, we now have multiple, always-on streams of information *about* media to parse such as Twitter, Flickr, blogs, check-ins, reviews, etc. We must treat our target as always moving and adopt technical methods which give us maximum flexibility in describing the content space.

The Four C's of Catalogs

Our goal is to identify each entertainment item (e.g. movie, TV episode, actor, etc.) with a numerical ID which allows each participant in the ecosystem to uniquely refer to the item throughout the distribution chain. The space of IDs is known as a *catalog*.

There are four attributes of a catalog ID space which we want to optimize:

- Coverage – How much content does the ID space address? The larger the

content space, the more useful the catalog.
- Cleanliness – How close does our ID space provide a 1-to-1 mapping from ID to real-world entity (i.e. no duplicates, no ambiguities). The cleaner the ID space, the better utility it provides the ecosystem.
- Churn – How quickly are existing IDs changing meaning[i] or becoming deprecated? We want to minimize churn in the catalog.
- Convenience – How easy is it for us to add new content to the catalog? What about new ―types‖ of metadata which we want to layer on top of the IDs?

Creating a catalog structure which optimizes these four tenets is difficult. The two obvious catalog construction techniques are: complete editorial control or fully automated construction. In an editorially controlled catalog, all new items are verified by a human gatekeeper before they are added to the catalog. In a fully automated system, a matching algorithm without human supervision is used to recognize uniquely novel entertainment works, and automatically creates a new ID and entry from them in the catalog.

These are how the two methods compare:

	Editorially Curated Catalog	Fully Automated Catalog
Coverage	Poor	Excellent
Cleanliness	Excellent	Poor
Churn	Excellent	Poor
Convenience	Poor	Excellent

In Section 4, we describe a hybrid method of catalog construction which helps overcome this contradiction.

Who Needs to use ID Catalogs?

From the perspective of a catalog maintainer, there are two groups of stakeholders in the ecosystem.
- Upstream Users. This group consists of content creators and owners who want to associate unique IDs with each of their entertainment works. In Section 5 we describe EIDR, a method to allow multiple upstream users to interact with a well-constructed catalog.
- Downstream Users. This group consists of content users (distributors, application developers, viewers, etc.) who want to access and describe entertainment content. These users may interact with multiple parties in the ecosystem and need to be assured that a common ID can be used cross-system and cross-device. In addition, downstream users want to be able to persist additional data (e.g. customer ratings, reviews, personal cloud

lockers, etc.) on top of the ID space and therefore need a stable catalog. In Section 4, we describe a catalog which has these properties and the methods used to keep a downstream ecosystem in sync as the catalog evolves over time.

Key for both groups of users is dealing with the constant cleansing and curating work a catalog maintainer such as Rovi conducts on the ID space. Two common issues which can cause problems are ID merging and splitting. As we describe above, the pace and variety of content creation and the need for a catalog with high coverage levels ensures that there will always be ID mistakes introduced into the catalog which are fixed post-entry. We define these operations as:

- Merges – When a single real-world entity is represented by more than one ID, we perform a merge to compress 2 through n IDs to a single ID.
- Split – When a single ID mistakenly represents more than one unique, real-world entity, we perform a split to turn a single ID into 2 through n separate IDs.

Since downstream consumers have a need to persist data on the IDs, we must invent a method to communicate merge and split operations throughout a distributed ecosystem. Our goal is to construct a methodology for the ID space which harnesses the contributions of upstream and downstream consumers but doesn't create chaos in the catalog. We describe this method in Section 3.

Multiple Stakeholders and Conflicting Design Goals

When trying to create a global unique identifier system that can be adopted on a global scale, one of the largest challenges in designing a unique identifier system is having a broad enough view of the entertainment industry to understand the needs of stakeholders at every step during an asset's lifecycle. Design pressures come from multiple, and potentially conflicting, business models, as well as individual stakeholder concerns around privacy and control. These issues are explored as part of Section 5.

SECTION 3 – TECHNICAL METHODS TO KEEP AN ECOSYSTEM SYNCHRONIZED

In a catalog ID space similar to that described in Section 2, we have upstream providers who want to register and contribute metadata to the catalog and downstream consumers who want to access the catalog and IDs to power applications. While we attempt to maintain a stable ID space, the dynamics of today's entertainment ecosystem ensure that cleaning, updating and curating are ongoing activities.

It is important to note that these activities occur simultaneously on both the IDs themselves and the underlying attributes of metadata associated with these IDs. For purposes of this paper, we will concern ourselves with the task of synchronizing the IDs themselves.[ii]

For downstream consumers, stable and synchronized IDs are required due to the need to persist data on top of the IDs. For example, assume you are an MSO providing a Video On Demand (VOD) service to your viewers. As your viewers watch VOD content, you want to be able to recommend them new movies and shows they may also like based on their previous viewing history (e.g. “You really liked *Meet the Fockers* last year and now *Little Fockers* is available on demand in HD”). In order to satisfy this use case, the MSO needs to store the household’s viewing history in a profile. The ideal method for accomplishing this is to keep a list of the IDs of video items previously purchased.

In an unstable ID space, it is possible that the ID for *Meet the Fockers* may “mean” something different now than it did when the MSO originally stored the data. This is a problem.
As described in Section 2, there are two situations where a change may occur to an ID post-creation: merges and splits. Ideally, a message could be broadcast to all downstream consumers whenever one of these actions occurs and the consumer could update their records. However, this approach is technically infeasible for several reasons:

- The ecosystem is fragmented and open. Previously, we could operate under the assumption that all downstream users were known subscribers by the catalog maintainer. However, in a world of interoperable APIs and dynamic data applications, we want to allow a large and distributed ecosystem to utilize the ID catalog without necessarily requiring centralized authorization.
- It is onerous to require downstream consumers to update records. If the MSO has stored IDs throughout their data systems, it is onerous to require them to propagate ID changes every time the catalog updates.
- It is brittle. If downstream consumers miss an update message, they will not be able to reconstitute their records and will have a corrupted ID space. Given the distributed nature of the ecosystem, we cannot rely on a synchronization protocol which is brittle.

An elegant solution which allows the resolution of splits and merges but avoids the problems described above is as follows: The catalog maintainer continuously monitors the ID space looking for errors. When the maintainer identifies two IDs referring to an identical entity which need to be merged, it:

1. Identifies the “dominant” ID. This is either the ID on which the most amount of activity has taken place or was the first to be created.
2. Creates a link from the subordinate ID to the dominant ID.
3. Stores the subordinate ID as a “deprecated” ID for the dominant ID.

Similarly, when the maintainer finds a single ID which erroneously refers to more than one real-world entity, it invokes a split mechanism which:

1. Identifies the “dominant” real-world entity, if possible. If not, selects the dominant as the first entity to be linked with this ID.
2. Creates new IDs for each entity.
3. Stores a link from the original ID to the two new IDs.

Now, how do we communicate these changes to the downstream consumers? First, we make the following assumptions:

- Downstream consumers have an incentive to know about ID updates.
- However, downstream consumers are not required to know about updates in order to operate. If they are unaware of an update and refuse to edit their stored data, the ID space should provide ID accuracy no worse than as if the update never occurred.

Given these assumptions, we adopt a system of implicit notification. Every time a consumer submits a request for metadata about a given ID, we include in the response the most up-to-date ID for this item. If no cleaning has occurred, this ID will be identical to the ID submitted. However, if there has been a change, the consumer will receive the new ID link. The consumer can choose whether to update their records and silently discard this new information. Since the catalog maintainer always knows both the previous meaning of each ID and the new meaning, the consumer will never run into ―dead IDs” which return no associated data.

An optional implementation includes an on-demand, ID lookup web service. In this implementation, the consumers can submit an ID and the maintainer will return its history and current state. This service is useful if the downstream consumer wants to check its records and perform larger-scale updates on its ID space data.

SECTION 4 – ROVI ID SPACES: A LOOK AT MULTIPLE ID CATALOGS

Now that we have a method to keep an ID space in sync throughout the ecosystem, how do we construct a catalog which maximizes coverage, cleanliness and convenience while minimizing churn?

Historically, there are two methods which could be used to maintain a metadata ID catalog:

1. Editorially controlled catalog. In this construction method, each addition of an ID to the catalog is controlled by a human editor. The human editor ensures that the ID does indeed represent a new, unique real-world entity. While this process maximizes catalog cleanliness, it is extremely resource intensive and slow to add new content (e.g. difficult to quickly grow coverage).
2. Fully automated catalog. In this construction method, a computer algorithm evaluates each new entity for uniqueness and then either matches it to an existing ID or makes a new entry to the catalog. This method makes it quick to ingest new content into the catalog. However, since it can be difficult to automate the matching of different sources of metadata, cleanliness will suffer with many duplicate ID entries.

Since neither catalog construction method is ideal for the purposes of creating a single entertainment catalog, we have developed a new, hybrid catalog construction model which allows for the dynamic aggregation of multiple metadata sources into a single ID space. We call this construction a Dynamic Aggregate Catalog (DAC). In addition, our method allows for both human and algorithmic cleaning methods and assumes these curation activities will be ongoing as opposed to only occurring on ingestion.

First, we describe our motivations for creating DAC. Then detail the methods underlying its use.

Over several years, Rovi has acquired multiple metadata databases including TV Guide, All Media Guide (AMG) and Muze. While each database describes overlapping content they do so for different purposes and in differing manners. For example, TV Guide had complied data on movies for display in a listings grid whereas AMG had longer descriptions geared for the retail setting. And, while TV Guide specialized in TV-related data, AMG also had deep information on television series provided they were made available in DVD boxsets for retail.

Even though these datasets had different conventions, our goal was to create a single, normalized catalog ID space which would allow us to access all available data. This is important because going forward, different types of applications will want data in different formats and optimized for different use cases. However, we do not want to create individual database silos locking away access to valuable information.

Rovi's motivation is not unique. As the pace and quantity of digital media consumption increases, the ecosystem will have access to multiple sources and catalog of entertainment content. They will need a way to organize and centralize this data around an ID space. Our DAC model can serve as a template for this process.

Inside the DAC

Figure 1 describes the DAC. The building blocks for the catalog are individual external catalogs of metadata such as AMG or Netflix. These catalogs can be growing (e.g. active) or fixed and can be either editorially controlled or fully automated. Each external catalog consists of individual items (represented by blue circles in Figure 1). External catalog items (ECI) have metadata layered on them. This metadata can be from metadata in the catalog itself or from 3rd-party sources persisted on the ID space itself.

Next, we create DAC items (orange) which link to an ECI. The DAC item inherits all source data from the ECI and as the source data grows and improves it is automatically synced with the DAC item. The data itself is stored referencing its original source. The link between a core DAC entity and an ECI has an inherent confidence value associated with it.

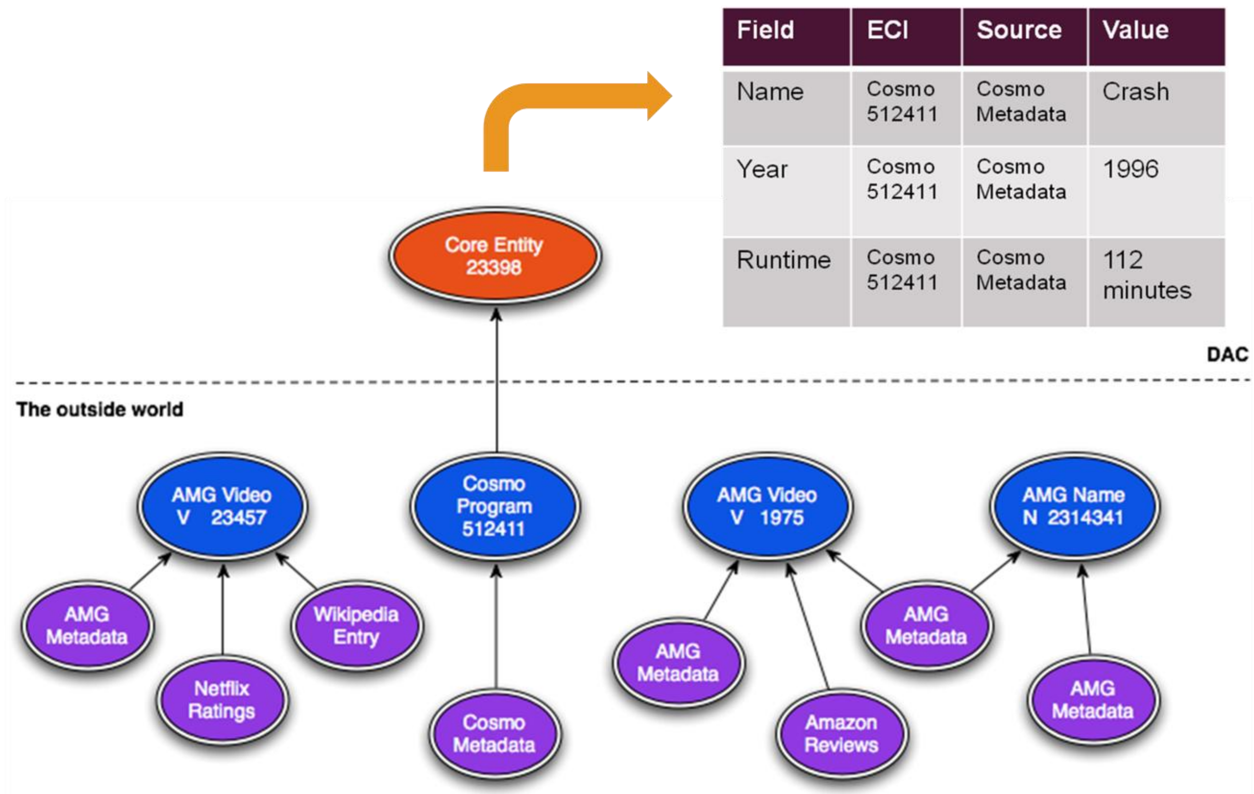

Figure 1: An overview of the DAC

Multiple ECIs may link to the same DAC item. We use software logic to reconcile conflicting values amongst the source data.

The DAC system allows for editorial curation **on top** of the catalog. The links between core entities and ECIs can be created automatically or via editorial means. When a link changes the underlying metadata follows automatically. This property is what renders the DAC a **dynamic** catalog.

SECTION 5 – THE ENTERTAINMENT IDENTIFIER REGISTRY (EIDR)

In section 3, we described the concerns of downstream consumers when interacting with an ID space. Simultaneously, upstream consumers have their own unique requirements for registering and identifying assets in the catalog. In this section we describe the EIDR system which meets these requirements.

In October 2010, a coalition of companies from the professional video ecosystem announced a new organization that would specifically address the creation of an identifier space for video assets. The organization, called the Entertainment Identifier Registry (EIDR), is a non-profit, centralized registry that was founded by companies such as Comcast, Disney, Warner Brothers, CableLabs, MovieLabs, and Rovi, and operates as an open organization for the standardization and adoption of EIDR.

The design of EIDR is based on another international standard and unique identifier system that has been around for over a decade – the Digital Object Identifier (DOI). This system uses a federated approach that links together unique identifier catalogs for widely disparate systems, ranging from identifying academic papers to uses for military applications to identifying video assets for EIDR. The core concept of DOI is to assign permanent unique identifiers that can be resolved using a URL-style scheme.

Figure 2 – Example of an EIDR unique identifier with full DOI notation

The DOI implementation for EIDR was developed by the Corporation for National Research Initiatives (CNRI) based on their open-source Handle System software. The modifications that EIDR made to the Handel System software embody a number of design trade-offs and considerations.

Opaque Identifiers

It should be noted that the identifier in Figure 2 is opaque – there is no information conveyed by the identifier itself about the asset that it is describing. The design tradeoff underlying this decision is one between permanence and human readability. While humans would like to use identifiers that they can read (such as ―doi/10.5240/Desperate Housewives/Season 1/Episode 1"), this only leads back to the reference-by-title confusion that led to the creation of identifier systems in the first place. Using an opaque identifier also means that any errors or changes, whether to spelling, hierarchy, or some other aspect of the underlying asset, will not result in changes to the identifier. This is key in enabling the identifier to remain permanent and unchanged throughout its life. The obvious downside is that opaque identifiers cannot replace labeling on physical assets that don't otherwise have distinguishing marks or labels on them.

Centralized Registry

One of the initial design decisions of EIDR was to create a centralized registry where each work is registered and assigned a unique identifier as part of a centralized repository. This design decision is a tradeoff of key business concerns such as accuracy, control, and privacy, where a centralized registry gains accuracy at the expense of the control and privacy of the organizations that are performing the registrations.

In order to realize the accuracy gains of a centralized registry, EIDR also employs a deduplication process that is a combination of automatic and manual comparison of records to determine the uniqueness of the record. The automatic comparison uses a scoring system to calculate the distance between two records and sets thresholds to determine whether the records should be considered the same or not. That distance is measured by the difference between the elements of the registered asset, which in turn had great influence on the definition of the metadata schema (as described below). There are cases where the distance between two records may be indeterminate as to their uniqueness, resulting in the need for human intervention to make a final determination.

To contrast this centralized model and its accuracy against the alternative, there exist other registries, such as ISRC for music, that use a distributed registry model. In a distributed model, multiple stakeholders receive an allocation of numbers and assign numbers to assets at their discretion. In a strongly distributed registry, the resolution of these numbers is also performed by the organizations that make the assignments; in a loosely distributed registry, the assignments are eventually communicated back to a centralized database without any checking to ensure the uniqueness of the records. In a strongly distributed registry, the organizations have ultimate control around the assignment of records and can determine when they are made public. In a loosely distributed registry, there is an opportunity to improve accuracy and reduce errors of multiple duplicate submissions through deduplication and other mechanisms; however, this comes at a cost of cleanliness from multiple erroneous records being published and having to be corrected after they are made public.

Metadata Schema

Another fundamental design decision for EIDR was the coverage of the registry, which is manifested through the definition of the schema. With the complexities of today's global reach of video assets and increasingly fragmented distribution channels, it was quickly decided that the registry should cover a hierarchy that includes three main groups: abstract, variations and encodings.

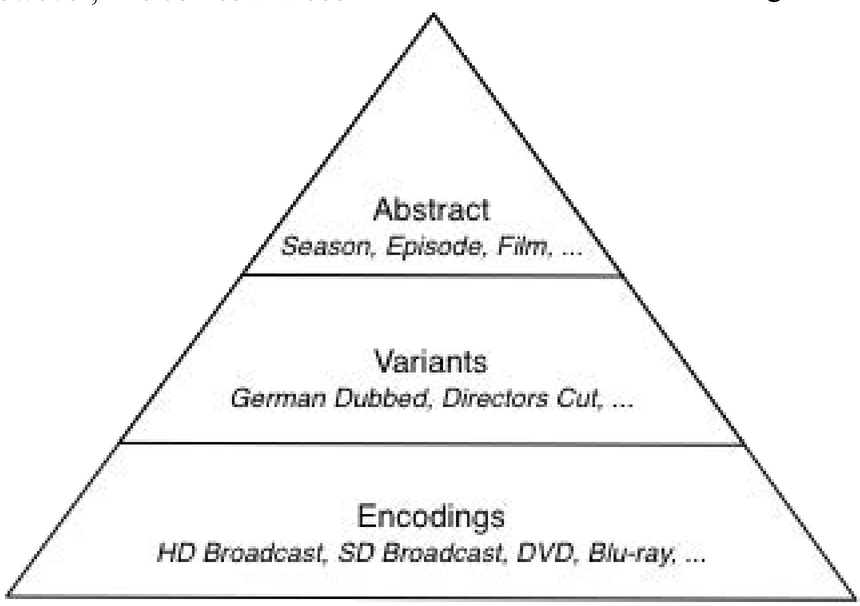

Figure 3 – Hierarchy of assets that are described by the EIDR metadata schema

Abstract records describe a group of assets at the ―title" level. These assets do not actually exist in one viewable form, but exist as multiple variations of a real asset. Title level assets include both series, which are a collection of television episodes, as well as episodes and films, which are typically a collection of variations of the video asset.

The variations are the next group down in the hierarchy and describe the different cuts and edits that happen during the post-production process. Examples include movies that have multiple edits, such as original theatrical release, directors cut, approved for broadcast television cut, approved for UK distribution cut (e.g. – no headbutts), safe for airline cut, the German dubbed version, the

French subtitled version, and so forth. Another example would be a television series, which is an abstract of multiple related television episodes.

At the lowest level of the hierarchy is encodings. EIDR has the capability to uniquely identify and describe each individual encoding of an asset, based on video codec, audio codec, codec resolutions and bitrates, and other encoding attributes. Examples include both digital media, such as a 1080 MPEG2 Transport Stream with a Dolby AC3 audio codec, and physical media, such as a DVD or Blu-ray.

Each group in the hierarchy has its own metadata schema and nomenclature used to describe that group of assets. While metadata is a vague and broad term, the design criteria for EIDR was to capture the necessary and sufficient fields required to ensure the uniqueness of each record, which ultimately supports the design goal of being able to provide deduplication of records across the entire registry.

A schema for EIDR in XML format can be found through the EIDR website: http://eidr.org.

Identifier Interoperability

A final design criterion came about through the desire for flexibility — in the realization that no single identifier system could have total and absolute coverage of all stakeholder concerns. To that end, EIDR included the ability to cross-reference other unique identifier systems. Each record in EIDR may reference one or more third-party unique identifiers. For the sake of near term design simplicity, these identifiers are treated as opaque objects that have few syntax restrictions and are not subject to the requirements around cleanliness (i.e. – deduplication) that the rest of the registry is subject to. Ultimately this gives EIDR a way of incorporating both existing standards and proprietary systems, as well as being able to expand through the cross-referencing of other unique identifier systems.

SECTION 6 – MSO APPLICATIONS AND ADOPTION

MSO Identifier Applications

Whereas the rest of this paper attempts to describe the formation and application of unique identifiers independent from individual stakeholders in the entertainment industry, this section will focus specifically on how unique identifiers can be applied to solve problems and enable new functionality and automation in MSO systems.

At a high level, the application of identifiers in MSO systems can be broken down into three categories:

1. Enabling the automation and distribution of video asset
2. Use in business systems surrounding the video assets
3. Use in providing features and functionalities that aren't currently widespread

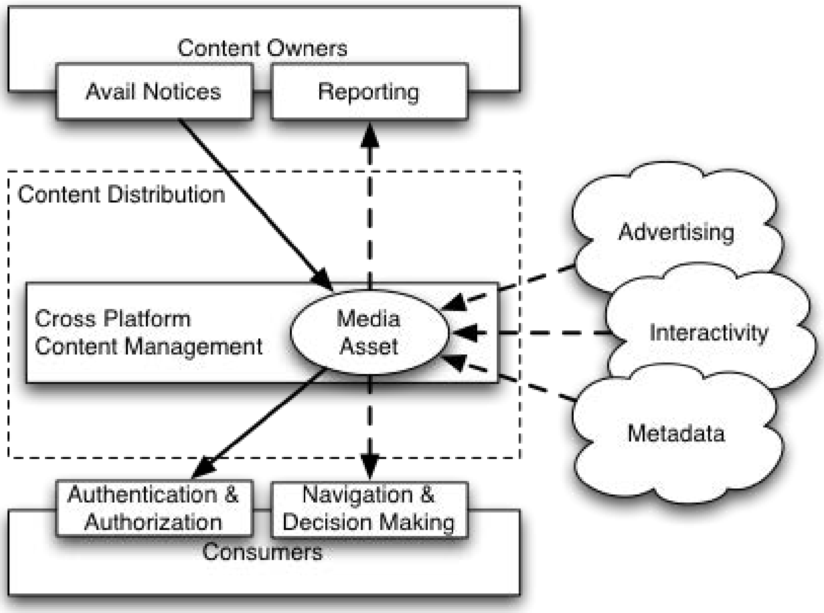

Figure 4 – Relationships between media assets and their uses that immediately benefit from a unique identifier system. Solid lines represent the applications of unique identifiers that directly facilitate the distribution of assets; dotted lines represent the application of unique identifiers in supporting business systems or in augmenting applications.

Asset Distribution

As studios and broadcasters integrate unique identifiers into their systems and move towards standard interfaces for the distribution of assets, unique identifiers will become ubiquitously available starting with the ingestion of the content. Until then, identifiers must be associated after content has been received, either through fingerprinting, watermarking, or metadata matching techniques.

Assuming that content has been received and an identifier has been associated with it, a content management system can then utilize the asset by associating it in any number of ways and especially to distribute it to the end consumer. In distributing it to the end consumer, the content management platform can use the unique identifier to check against distribution rules (such as the rights to distribute to various platforms, or the windows of when content is available in specific distribution channels) or to check a consumer's right to access a specific piece of content as part of a subscription.

Business Systems

Unique identifiers also play a role in the supporting business infrastructure, including finance and auditing systems. For example, when a consumer watches a VOD asset the unique ID of that asset may be transmitted back to the finance reporting system along with other billing information. Finance departments can then use this unique identifier for their own internal auditing, or the number may be shared with other departments that are responsible for sharing consumption information with the content owner.

Another common application of unique identifiers in business systems is for antipiracy. The unique identifier may be embedded into assets as an audio or video watermark, which enables forensic teams to recover information about which asset was pirated (for example, which edit or which encoding).

Augmenting Applications

The final and most interesting use of unique identifiers is their application to emerging technologies that require scale and automation beyond what can be done with a manual labor force.

One example of this is advertising. As the number of video assets and the platforms that they are viewed on both increase, so do the number of advertisements that can be placed. This creates both a larger market for advertising and a problem of cross-platform measurement. In order to solve the challenges around placing ads and aggregating the viewership metrics to be reported back to the ad buyers, a common unique ID system must be established between all stakeholders.

Along the same lines, interactive assets such as EBIF are becoming more common for both augmenting consumer experiences and creating more engaging advertising with new calls to action. While interactive assets are sometimes embedded into video streams by broadcasters, they may also take a peripheral route, requiring that they be associated and reattached with intended video asset(s).

Another class of problems is presented by the metadata associated with video assets. Both schedule and program data are delivered by third-party sources to the MSOs, requiring that they be re-associated with the broadcast programming and VOD assets in order to assist with both content management and ultimately with the navigation and decision making experiences at the hands of the consumer.

Adoption

The ability to adopt a unique identifier varies based on the criteria of each application. While some applications are highly dependent on the ―four C's" described in Section 2, others may have secondary adoption criteria. Those adoption criteria include:

- Number of assets that are currently identified, as related to the total population of assets
- Legal terms of use or intellectual property restrictions around the use and propagation of the identifiers
- Network effects from adoption by other applications, systems, or actors
- Ability to associate identifiers with assets

---

[i] What does an ID ―mean"? For our purposes, an ID should map to a real-world entity. For example, the AMG ID "V50435" refers to the original release of the movie *Top Gun*. As the ID space is updated, we always want this ID to point to the movie *Top Gun* which is something people in the real-world can watch, purchase, and have opinions.

[ii] Ensuing the ecosystem has access to the most accurate and clean metadata is, of course, also an important design goal. However, the technical challenge is mitigated once the ID space issue is removed. Once a downstream consumer has access to a stable ID, they can access on-demand services (e.g. a RESTful metadata service) to obtain the freshest possible metadata or receive regularly scheduled updates in bulk.