Content Recognition Rules

Email comments and questions to CRR@movielabs.com

Notices

CONTENTS

1 INTRODUCTION

1.1 Problem Statement

The management of intellectual property rights around User Generated Content (UGC) is critically important if video on the web is to move forward creatively and commercially. This document provides a technology framework for dealing rules (whatever they might be) to manage intellectual property rights within the high-volume, rapidly evolving technological and commercial reality of the UGC community.

Content recognition systems from a variety of vendors are starting to work quite well, and many ISPs, UGC sites, and universities are beginning to deploy them to detect content that meets someone's notion of infringement of someone's notion of rights. Both the infringement and the consequences in existing systems tend to be very black and white – something infringes, or it doesn't, and if it does, the content is taken down. For example, the Automated Copyright Notice System (ACNS), found at http://mpto.unistudios.com/xml/, provides a DMCA-compliant way of communicating a single action – the removal of the offending content.

This simple yes/no decision is not really a satisfactory solution. Different content owners and rights holders will (and already do) have different ideas of what constitutes infringement and permitted use, and legal opinions about what constitutes fair use or permitted use will continue to evolve. Both Content Owners and UGC sites are very interested in forming partnerships where permitted and licensed use of recognized content are the desired solution.

Currently, there is no common way to express a set of rules, or even concepts, about what to do when a piece of content is 'recognized.' Expressing and communicating these rules is implicit in the statements from the sponsors of the *UGC Principles* and other sources (see section 1.4 below.) Without a vehicle for expression and communication, confusion and inconsistency will prevail, and progress will be slowed because of fear, inconsistency, and an inability to change with a complex and dynamic market. The following framework provides an appropriate mechanism for expressing the rules for managing the use of content between content owners and UGC sites.

1.2 Mechanism, Not Policy

This document does not dictate, or even propose, a definitive set of rules. Rather, we want to provide a mechanism that allows the expression and communication of a wide variety of rules, applicable to a large number of situations. Without such a mechanism, it will be impossibly difficult to monetize, experiment, learn, and adapt as technology, distribution, and the law change.

Our key concerns have been:

- Clarity – it should be almost impossible to misinterpret what a rule means or how to apply it in good faith

- Implementation – this is a current issue, not a future issue; any solution must be feasible now, using current technology
- Precision – there are many nuances around fair use, permitted use, and infringement, which have to be covered (subject to the previous two items)
- Automation – the volume of video content in the world increases ever more rapidly, and has already outstripped the ability of humans to cope with it
- Exception handling – it must be the case that human skill and reason can be brought into the process when needed; we hope this will be infrequent, but it must be allowed for

Two traditional design criteria have received less attention. Extensibility falls out almost automatically from using XML, and completeness has taken a back seat to the 80/20 rule in order to serve the goal of a near-term implementation. (For items considered or suggested that did not make it into this specification, see the appendix on future features.)

In order to support the goal of having implementations sooner rather than later, the specification allows for both full and baseline implementations. The baseline is a strict subset of the full feature set, and includes a simple set of detection criteria and actions; the full version incorporates more complex criteria and actions, as well as mechanism to support time-based windowing. To avoid confusion and version chaos, we strongly recommend that an implementation be one or the other, and not something in between.

1.3 Executive Overview

We believe that it is necessary (and possible!) to develop a way of expressing an interestingly large set of rules around content recognition and its consequences using very simple syntax. We have chosen to use XML, though RDF and OWL are also viable candidates.

These rules can express a variety of notions of 'infringement', based on length, percentage of the UGC content matched, percentage of an original work matched, and so on, and a variety of actions to take, including 'take it down', 'leave it up', 'surround it with advertising', etc., all modifiable by time windows and geographical constraints. Permitted use is in some ways the flip-side of infringement, and so has equivalent flexibility.

The rules, criteria, and actions that are expressible in this format are independent of the underlying content recognition system – they are equally applicable to watermarking systems and fingerprinting systems.

Some examples of actions that can be expressed by this framework are:

a. On recognition of at least 60 seconds (for example) of this asset, please remove it from use.
b. This asset is playable only in the US.
c. This asset is not playable in the UK until July 4, 2008.

d. In a mashup of multiple assets from the same series, if the total time of all assets from the series totals 3 minutes, then remove it from use.

e. On appearance of this asset on a UGC site, send an email notification to the rights holder.

f. When delivering an uploaded copy of this asset to a consumer, some ads are associated with it and should be shown.

g. If an uploaded video contains over 60 seconds from this movie, and that represents over 50% of the video's total length, quarantine it, pending investigation.

h. If an uploaded video contains more than 33% of an original asset, take it down and notify the originator of the copy and the owner of the original.

i. If an uploaded video contains an AACS theatrical use only watermark, send a DMCA notice.

j. If the quality of an uploaded video is low enough, take no action.

k. If the last 3 minutes of this show are found in uploaded content, replace the UGC with a teaser clip.

There are more detailed examples of rules and their implementations in the Use Cases section of this document.

An asset can have multiple rules associated with it, which can vary with the location of the uploader or the location of an individual attempting to access the copy.

A content owner may have different rules for different sites or ISPs, in which case each such entity would receive a different set of rules. We expect that the XML will be generated using a UI- or database-driven tool rather than by hand. Supporting multiple targets in such a tool is a simpler task than specifying and supporting multiple recipients in the same container. Therefore, the current version of this specification supports one set of rules per XML document. (See section 6.9 for further discussion of this.)

So, although this document uses the phrase 'the rules for an asset', that in no way implies a single set of rules for each asset, and should be read as 'rules for an asset for a particular site or ISP.' The separation of rules into a separate file for each recipient also makes it easier to keep per-recipient information confidential.

This document does not imply any sort of policy or specific actions to take, but provides the mechanism to express a wide variety of such policies. The syntax defined is normative, or prescriptive and definitional; the examples and use cases are informative – just explanatory content given for the sake of clarifying the use of the syntax.

The framework has been developed with input from content owners, vendors of content recognition systems, ISPs, UGC sites, and industry bodies.

1.4 The Principles for User Generated Content Services

A group of content owners and UGC sites have announced "The Principles for User Generated Content Services" (hereafter *UGCPrinciples*; please see the References section for

more details) which sets out a consistent and coherent view of the legal, moral, and commercial rules of the road for this area. *UGCPrinciples*, however, leaves open two important questions:

- What does it mean to match a piece of content, and how is actionable infringement of the rights of the copyright holder determined? The rules will need to be different for different content owners and also the type of media asset (e.g., movies vs. newscasts.)
- *UGC Principles* §3.a.2 states that a Copyright Owner needs to have "provided… instructions regarding how matches should be treated." This is only possible if there is a clear and unambiguous way of providing the instructions.

This document tries to address these issues and provide a framework to allow content owners a way to express the actions they want taken for a specific media asset.

Another view of the subject is taken by the Electronic Frontier Foundation (EFF) in "Fair Use Principles for User Generated Video Content" (hereafter *EFF Fair Use*; please see the References section for more details). *EFF Fair Use* sets out a different set of rules but similarly to *UGC Principles*, it does not define any technology for communicating the information.

The rules described by *UGC Principles* and *EFF Fair Use* can both be built on top of a common technology framework. With the exception of the 'license' action in *UGC Principles*, they can both be implemented on the baseline version of this specification, as can many other policies. However, both systems also need an operational framework, for dealing with take-down notification, appeals, legal responsibilities, and so on. That operational process is outside the scope of this document, which covers only the communication of rules and the actions that are their consequences.

Based on input from copyright owners, service providers, and providers of content recognition systems, the description here is sufficient for creation, distribution, and implementation of a set of precise rules, covering:

- Criteria that apply once a basic match has been determined and are evaluated before triggering any actions; these can be used to implement the filters in *EFF Fair Use* §2a, as well as many others.
- The actions 'remove', 'allow', and 'license' from *UGCPrinciples* §3.c and replace from *UGCPrinciples* §8, as well as other common actions
- A format for communicating the actions and the triggering criteria to systems that implement the actions when content is uploaded or accessed

2 DOCUMENT ORGANIZATION AND BACKGROUND INFORMATION

An architectural overview is presented in Section 3.

The system is first described by a set of tabular definitions with simple examples in XML for both specifying the rules and communicating the actions. This is found in Section 4.

Section 5 describes the required features for a baseline implementation and a full implementation.

Section 6 has a set of complete examples for some use cases; these examples also illustrate best practices for some common situations.

Section 7 contains information about the XML schemas for the rules and notifications.

Section 8 is a list of reference documents.

Appendix I is a copy of a short patent license applicable to this specification.

Appendix II is a set of possibilities for future extensions.

2.1 Notational conventions

In the specification section:

Time, Date, DateTime, Duration, Integer, Boolean, URL, and String are assumed to be the standard XML types, and are not otherwise defined in this document. See the References section.

Time, Date, and DateTime must include a timezone. If one is not present, an implementation is free to flag it as an error or assume Z.

Attribute and element names in *italics* are optional in syntactically and semantically valid XML.

Attribute and element names in non-italics are required for syntactically and semantically correct XML.

Element names start with a majuscule, and have inter-caps, e.g. Element, BigRequiredElement, *Extra*, or *OptionalElement.* Anything that does not adhere to this is an error, and should be reported to the document editor.

Attribute names start with a minuscule, and have inter-caps subsequently as needed, e.g. attribute, attributeWithLongName, *extra*, or *optionalAttribute.* Anything that does not adhere to this is an error, and should be reported to the document editor.

'Multiple values may be specified' and similar language mean that multiple instances of the element in question are allowed, rather than, say, a comma-separated list in one element.

2.2 Definitions

AACS – Advanced Access Content System, which defines standards for various kinds of copy protection. See references.

ACNS – Automated Copyright Notice System, a format for communicating notice of potential copyright infringement to owners and originators. See References.

Asset – A unit of content or a piece of media. See Site Asset and Original Asset.

Candidate Asset – A site asset that is being checked by an identification or recognition system for similarity to a set of reference assets.

Charybdis – see Scylla.

Coral – A set of DRM interoperability specifications. See References section.

DCI – Digital Cinema Initiative, an industry standard for distribution and protection of digital theatrical content. See Reference section.

Fingerprint—A set of bits generated from an asset such that fingerprints generated from an unknown asset can be used to associate the unknown asset with one or more original assets by comparing the fingerprint from the unknown asset to a reference database of fingerprints from original assets.

Group – A set of original assets that are related somehow; for example, all the episodes of a series might form a group.

Identifier – A sequence of bits, usually represented as a number or string, used as reliable shorthand in a particular context for referencing a set of information. Examples include database keys, product SKUs, and email addresses.

ISAN -- International Standard Audiovisual Number, a standard identifier for audiovisual information. See References section.

Master Asset – See Original Asset.

Original Asset – An asset that has been entered into the reference database (q.v.), sometimes called a reference asset. This is an asset against which uploaded content will be compared.

Originator – The source of a site asset; the person (ideally) or IP address (at least) that provided a particular site asset.

Owner – An entity with the authority to provide a set of rules for an Original Asset. The term is used loosely, rather than in a strict legal sense, and it is up to the entity ingesting the rules to determine whether or not the sender of the rules is allowed to submit them for a particular piece of content; resolution of any conflicts on that front is beyond the scope of this document. Often the 'owner' will be a copyright holder or distributor, but it can also be some other authorized third party. For example a law enforcement agency may request that terrorist or pornographic videos be entered in the reference database along with associated rules.

Reference Database – Information about a set of original assets. See the 'System Model' section for details of its construction and use.

Rule Instance – A rule that is directly associated with an asset; it is the asset's own copy of the rule, and can be changed only by replacing it. When an instance rule changes, only the asset to whih it is attached is affected.

Rule Template – A rule that is associated with an asset by reference. When the rule template is changed, all assets that refer to it use the new rule. The connection between asset and template

can be broken by making the asset refer to a different template or by replacing the template with an instance.

Scylla – see Charybdis.

Site – Loosely used to refer to the external manifestation of a collection of hardware, software, and assets, such as a UGC site or a file-sharing site. All sites will have IP addresses, and most will have domain names as well.

Site Asset – An asset at a particular site, such as an uploaded video at a UGC site.

UGC – User Generated Content, conventionally referring to uploaded audio and video assets

URI – Uniform Resource Identifier

URL – Uniform Resource Locator; in this document it is used as shorthand for a URI with http as the URI scheme.

UUID – Universally Unique Identifier; 'universally unique' is often interpreted as 'highly probably unique'. See References section.

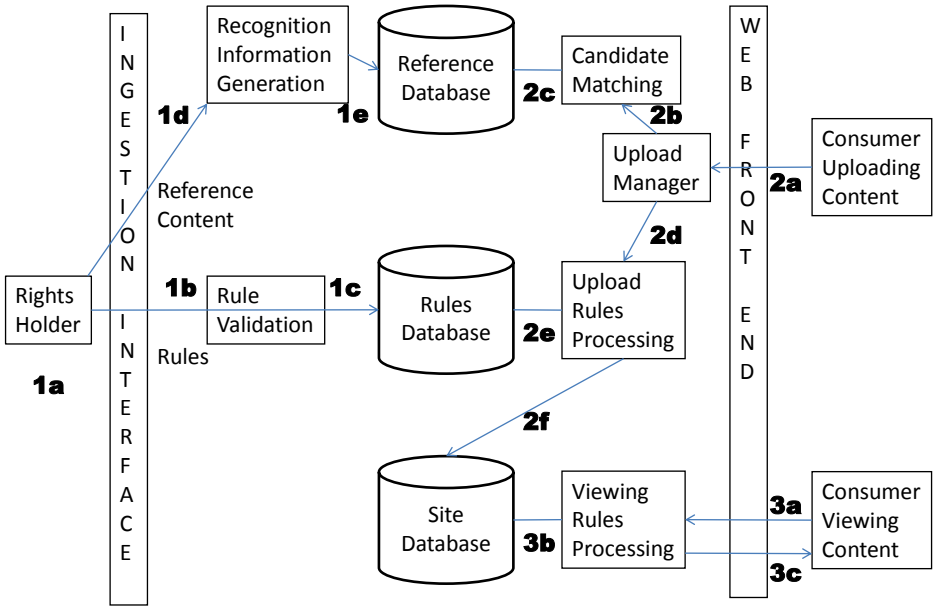
Watermark – An identifier that can be added to an asset for later extraction. Ideal watermarks in the context of content recognition systems generally are non-removable, imperceptible to human senses, and immune to standard AV transformations.

Whitelist – A whitelist is a mechanism used to explicitly and efficiently allow a known party to do a particular thing, bypassing usual processes. For example, content submitted from the marketing department of a studio might be accepted onto a UGC site with no further checking or questioning. The opposite of a whitelist is a blacklist, which explicitly and efficiently disallows a known entity from doing something. Whitelists are used when the number of exceptional allowances is small, blacklists when the number of exceptional prohibited parties is small.

3 SYSTEM MODEL

3.1 Logical Components

The logical model for the system is as follows:



First, a database of reference content and rules is generated, the contents of which can change over time as rights holders add or remove content or change rules for existing content.

1a: Rules and a reference asset are sent to an ingestion system. The rules describe criteria that refine the concept of 'matching' and actions to take if those detailed criteria are met. Some of the actions are taken when a consumer uploads content and some are taken when a consumer attempts to view uploaded content. It is expected that the XML for the rules will be machine-generated (via a GUI or some other tool) rather than hand-built.

The rules and reference asset can come at different times, from different sources. They must have the same OriginalAssetID.

1b. Validity checking is done on the rules. This can include, for example, verification of syntax and structure and determination of conflict with rules already in the system.

1c. After the rules are validated, they are stored in a Rules Database. They can be stored as the original XML, or processed or compiled into a form that better suits the runtime environment.

1d: The reference content recognition information for the reference asset, such as fingerprints, metadata, and watermarks, is generated. This step can be performed earlier

in the process (e.g. by or on behalf of the rights holder), in which case only the reference content recognition information will be sent in step 1a, not the reference asset itself. Reference information for audio and video components may be generated separately.

1e. The reference content recognition information is stored in a database along with the rules for the asset. Reference information for audio and video components may be stored separately.

When someone tries to upload content, the candidate content is compared with assets in the reference database and actions are taken if the candidate content is recognized.

2a: The candidate content is sent to a front-end upload management system

2b: The upload manager sends the candidate asset to the recognition system.

2c: The recognition system checks the candidate against the reference database or databases, and returns information about any matches to the upload manager.

2d: If there is a raw match, the match information is sent to a rules processing system.

2e : The rules associated with the reference assets are fetched from the rules database and evaluated as follows for each matched reference asset:

- From the rule lists for any matched reference assets, the system determines the highest priority rule or rules the detailed criteria of which have been met as a result of the identification.
- For each such rule, the associated actions are communicated to the systems that perform them. It is possible that multiple rules will apply from multiple reference assets and that multiple actions will be triggered.

2f: Allowable content and any use-time actions associated with the more detailed matching are delivered to the database of content on the site.

When a consumer tries to access uploaded and authorized content, any use-time actions associated with the content are performed.

Step 3a: The consumer requests the content through the front-end system

Step 3b: Access-time actions associated with the content are fetched from the site database, filtered, and run.

Step 3c: The results of those actions are used to determine the content that is returned to the consumer.

3.2 Underlying Systems Requirements

3.2.1 Recognition Systems

To support the baseline specification, the minimal requirements from a content recognition system are:

- The ability to return an indication of the length of a reference asset
- The ability to return an indication of the amount of content in a reference asset matched between a site asset and a reference asset
- An indication of the component to which the match applies – audio, video, or both.
- The ability to return the above information for all reference assets to which the site asset is successfully matched

To support the full specification, the underlying system should be able to return:

- The length and position of a segment matched in a reference asset. The resolution of the time and length are system dependent , but the system should make a best effort attempt to meet the 1-second granularity of the specification
- The abstract quality or sureness of a match, normalized to a linear scale of 0 – 100. A system that can only return yes or no matches should return 100 and 0.

Beyond that, details of matching systems are outside the scope of this document; some general overviews of the subject are given in the References section. However there are some caveats which, taken together, demonstrate the inadvisability of assuming matches at the level of an individual frame or perhaps even a single second. For example:

There are many nuances in the step of 'recognizing a media asset', and this requires navigating between the Scylla of false identifications and the Charybdis of missed matches. Recognition and matching algorithms differ in the length of the content needed to attempt a match, the number or duration of individual matches needed to make a positive identification, and the reliability of identification.

There are also variations in what 'match' means. As a very simple example, consider the case of adding blank frames between frames of the original – if there is one only every 5 seconds, most systems would agree that there is a match, but if there is one between every pair of frames of the original, should it still match?

Finally, some of the rules work in terms of length of content matched either explicitly (as a time) or implicitly (as a percentage.) Recognition systems can be fuzzy around the concept of length matched; using the example above, interspersed blank frames may or may not count towards the match, depending on where in the matching process the bookkeeping is done. Reference assets may have blank sections at the start or end as artifacts of the production or encoding process, which will impact computation of percentages.

3.2.2 Content Sites

In order to meet the guidelines in *UGC Principles*, content sites should implement a whitelist for allowing authorized or privileged uploads. The management of that whitelist could be site-wide, per asset, or per rights holder.

Content sites also are assumed to have means for logging and auditing decisions and actions and emailing notifications to rights holders and consumers. These are outside the scope of this document.

It is not intended that reversal of takedown notices use the communication mechanism described here, and so such action is also outside the scope of this specification.

3.3 Implementation Considerations

There is enough flexibility in the logical architecture to allow many possible implementations, even when the recognition rules are added into pre-existing systems. Content recognition systems and UGC sites are already individually complex, as are the interactions between them. Implementation concerns raised by this specification generally occur where those two macro-components interact with each other differently because of the addition of the rules mechanisms; or where pre-existing implementation decisions raise constraints around database design or connectivity across components.

It should be noted that there are two systems not shown in the above diagram. One is a mechanism to dispatch actions to the systems that execute them, and the other is a system that maps among various possible ID domains (site-specific IDs, owner-specific IDs in content rules, and IDs used by recognition systems.) Both of these are simple and in all probability already exist in most systems; they are included here for completeness, rather than in the expectation of having implementation issues.

Even though the number of components in the architecture is large, there are a few ways to group them (logically and even physically) that are representative of the way recognition systems and UGC sites tend to work now and plan to work in the future.

For these descriptions, 'local' and 'external' mean 'local/external to the UGC site.'

1. The content matching system with the reference database and the upload rules processing with the rules database are external; the exact actions to take when content is viewed are returned by the external system, probably as the XML-defined Action object defined in this document, are placed in the site database (perhaps after some pre-processing to meet the site's implementation needs) and used by the viewing rules processor. This model is similar to that used by some service providers today.
2. The content matching system (or, less commonly, just the reference database) is external. The upload manager hands the result of the external matching process to the local upload-time rules processor, which applies rules from the local rules database. Steps subsequent to determining the Action to take are as in Case 1. This model is used when a UGC site outsources the matching activity to an external agent.
3. All systems are local. This allows for tighter coupling, perhaps for reasons of control, performance, or coping with legacy systems; the viewing-time rules could be kept in the rules database rather than in the site database. Tight coupling may also make it easier to implement advanced features that are allowed (but not required) by this specification, such as holding an uploaded asset when an embargo for it is about to expire.

There are many other ways of laying out the components, of course.

3.3.1 Ingestion of Reference Assets and Rules

This document purposely does not define a transport mechanism for the XML description of the rules. Systems that already take data from content owners can add the rules to an existing data feed or add an additional one; other sites can collect the rules via ftp, RSS, or any other means. The communication mechanism should be secure and authenticated; it must also have a way of returning status.

Some vendors of content recognition systems maintain recognition databases themselves and provide their customers with access to them via either a local copy or remote communication, and some recognition systems generate and store the recognition information at the site where it will be used. Both of these schemes have commercial and technical merits and disadvantages; the system architecture accommodates both of them.

The parts of the rules used when content is recognized and uploaded and the parts used when successfully uploaded content is accessed are logically separate, giving the system three logical data repositories.

These three repositories could be in 1, 2, or 3 data stores; this decision will be partly based on which of the 3 cases described in section 3.3 is used. There are some other considerations as well -- a system with a single database is unlikely, since it is highly probable that the performance tuning needs of current recognition system require the recognition data to be stored in its own format in its own systems. Keeping the recognition rules and the actions for content access together or separate also depends on a variety of factors; even in cases where they could be kept together (such as 1 and 3 above), which has the virtue if simplicity, there might be performance requirements at access time that argue for separating them.

3.3.2 Content Upload and Recognition

Communication from the upload system to the recognition system is outside the scope of this document. Systems that currently return or receive a Boolean decision (recognized/not recognized) will have to be extended to handle somewhat more information.

There are several ways of executing the rules for a piece of recognized content, depending on the recognition system, the database arrangements, and the communication paths available. It is expected that vendors of recognition systems will provide integration interfaces that allow for closer or looser coupling of their systems to the UGC site for reasons that are either commercial (e.g. providing a service vs providing licensed software) or technical (e.g. gaining access to site-specific data.)

When a candidate asset is accepted by the UGC site, the actions associated with the rules that it matched have to be associated with the asset and stored for access by the system that applies rules when content is accessed. See the previous section for a discussion of possibilities for data repositories.

Finally, there may be some performance advantages to storing these access-time actions in a non-XML form that is simpler for the content serving system to parse and use.

3.3.3 Content Download to Consumers

The process for providing content for viewing is what one would expect on a site, with the addition of applying the actions associated with an individual site asset. It requires a new mechanism added to the site's infrastructure if the actions are stored separately from other information the site stores about the asset (e.g. asset name, ID of the person who uploaded it, and so on.)

The evaluation and execution of the actions can be tightly or loosely coupled with the rest of the site's content serving system, with the same kinds of considerations as given above for execution of rules at upload time.

3.3.4 Performance Analysis

The rules can add load to a system at all three of the points given above. Here we consider the performance impact of implementing this specification compared with performing content recognition with a fixed "leave up" or "take down" outcome.

3.3.4.1 Ingestion of Reference Assets

There is extra data traffic at ingestion time – the XML rules – but this is tiny compared to the size of even a short video.

There is some extra computational work to decide where and how to store the rules (whether they are in one database or two), but that is once again tiny compared to the computational load of generating information for the recognition database.

Thus, an implementation of this specification should add no new bottlenecks at reference content ingestion time.

3.3.4.2 Content Upload and Recognition

There is no extra data traffic at upload time, but there are two ways in which the recognition-time rules can add load to the system.

There is extra processing after content is recognized that must be serialized after the recognition system runs. This involves parsing the rules, applying the criteria, determining which action to take, and taking the actions. The computational load for this is expected to be small relative to the work needed to do the initial recognition. Some rules may require adding access-time actions to a piece of content, but that will add only a small amount of data to the records that are kept when content is accepted into the system.

Some content recognition systems stop processing once a single match or part of a single match is found. This is fine for simple take down/leave up decisions, but inadequate for more sophisticated business models and rules. Although the baseline does not, the full specification requires the recognition system to run more fully to support aggregate matches and the execution of rules from multiple rights holders, as in the case of mashups of sources from different owners. The baseline and full specifications both require that the recognition system return the amount of an asset that is matched, rather than a simple yes/no, which likewise precludes certain short-

circuiting optimizations. In both cases, this extra computation would be an artifact of any system that supported finer-grained business models, and so is not unique to this specification.

Thus, there are not expected to be any performance constraints at upload time that would not also exist in any other system that supports more sophisticated and emerging business rules.

3.3.4.3 Content Download to Consumers

Extra load can be added to the system if there are actions to perform when a consumer requests the content. The actions allowed fall broadly into ad insertion and availability (temporal and geographic.) In all cases the rules can be pre-processed into the most efficient form for the system that executes them.

Most pages at modern web sites are already dynamically generated, so adding or increasing dynamic ad insertion doesn't complicate things much. If this really does become a concern (which we expect it won't), certain kinds of ads (text, banner, pre-roll video) can start to be gathered as soon as a page is viewed, although at the cost of significant added complexity.

Temporal filtering to determine the action to take is a database lookup followed by a comparison, and so should not be expensive. If for some reason it is expensive, then actions that are currently temporally valid can be generated from a master repository and stored into the database from which the site works; expired actions would get updated with original actions filtered from the master repository.

Geofiltering is still a maturing technology, but it is already deployed at many video-oriented sites so far has not produced any noticeable performance degradation.

As with the other two areas, it is expected that adding the recognition rules will not impose any noticeable performance penalty or computational load when consumers access video content. In the unlikely event that there are issues, there are well-known techniques for caching and pre-computation to ameliorate them.

3.3.5 Security Requirements

The ingestion of reference assets and rules should use a secure mechanism, since the rules are commercially sensitive information, and the reference asset is valuable in its own right.

Notifications and logging to content owners will ideally be done over a secure channel, but some systems may start off using ordinary unencrypted email.

In a loosely coupled implementation, communication among remote systems should be over a secure VPN, TLS, SSL, or the like.

Finally, the recognition rules should be treated as 'commercial in confidence' or trade secrets, with appropriate measures in place to secure the database or databases in which they reside.

3.3.6 Error Handling

The ingestion process needs to be able to return status; errors can occur when receiving, parsing, or saving the rules; errors can also occur during generation of recognition data for a reference asset. If there is an error ingesting a rule file, the whole file should be rejected.

Although this seems draconian, it keeps systems from getting into unknown states, which can be hard to sort out after the fact.

Most of the errors at upload time will be standard web application errors and outside the scope of this document. If the recognition system is unavailable when content is uploaded, it can either refuse the upload or accept it and queue it for later analysis; in the latter case it is preferred that the site not make the content available until the analysis is done, but that is not mandatory.

Most errors when site assets are accessed will also be standard web application errors, and so outside the scope of this document. If the system that processes access-time rules is unavailable for a short time, it is probably not realistic to prohibit access to the content – the adverse effect on consumers would be high and the damage to content owners relatively low. If that system is unavailable for an extended period, or a commercially significant amount of time, then prohibiting access becomes more viable.

3.3.7 Conflict Handling

Even though individual rule sets may have no errors, they can still be in conflict with other rule sets. Some of these can be handled in an automated way, but many will to be handed over for resolution by humans based on contracts, business practices, operations workflow, and so on. This section covers cases where things can be resolved technologically and provides some guidance for cases where they cannot.

3.3.7.1 Ingestion-time conflicts

The most obvious case here is having multiple owners provide rules for the same reference asset (where ‘same’ means having the same Original Asset ID.) In that case, if the owner fields of the rule sets have non-conflicting geography details, there is no conflict, and rules should be applied based on the geography information. If there is conflict in the geographies, then manual intervention is required. This will take place at rule ingestion time, and so the delay in processing that this causes ought to be acceptable.

Entities that accept rules from owners must specify their process for this resolution. A reasonable starting point for this process is:

- A set of rules already ingested takes precedence
- All parties involved in the conflict are notified

3.3.7.2 Upload-time conflicts

When content is uploaded it might trigger actions from multiple assets if more than one asset is matched. Trailers are a special case, and can have special treatment – see the ‘Managing Trailers’ section of this document. Other cases in which this can happen include:

- A mashup has content from more than one reference asset; in this case the assets are completely different and the owners may well be too.
- An uploaded item matches multiple similar reference assets, for example the theatrical release of a movie and the TV release of the same movie.

- An identical reference asset has been provided under different IDs, perhaps from different rights holders.

In all such cases, all content owners involved should be notified, for the sake of courtesy and clarity. Fully automated resolution is not possible, but reasonable steps to follow are:

- The actions from the rule with the highest priority should be executed.
- If there are multiple highest priority rules from different assets, the one with the highest quality of match should be executed.
- If the match quality doesn't disambiguate, then a TakeDown action, if present, takes priority over any other actions.
- Otherwise, the Quarantine action should be taken, and the owners of the reference assets contacted for resolution.

Finally, in case of a conflict or a match of suspect accuracy, the implementer always has the option of flagging the content for manual review.

3.3.7.3 Access-time conflicts

If the above guidelines are followed, there should be no access-time conflicts. However, it may be the case that ad revenue should be shared among multiple entities, for example when a conflict between content owners is resolved by an agreement to make the content ad-supported. This case is included for consideration in some of the possible future features.

3.4 Timing, Expiry, and Updates

Any business system that uses time or timed events will have to reconcile the intent of the commercial rules with implementation and operational constraints. This section provides suggestions and clarifications for some of those areas. This particular set of issues will require some patience from all parties, and any real-world experience obtained should be folded into the next version of this specification.

3.4.1 Expiration Times

The expiry time of a Rule is the earlier of the expiry of the RuleList itself and the expiry of the Rule.

Any actions that have been cached with a matched site asset should be cleared from the system at expiry time. In an ideal implementation this would be perfectly precise, but it is understood that large sites may have a delay in doing so, even when using optimized reapers or garbage collectors.

3.4.2 Rule Update

When a set of rules is ingested for an asset that already has a set of rules, it is to replace the entire existing set of rules for that asset, following this constraint:

- If the new rule has no start time, the replacement occurs as immediately as possible; otherwise, it take effect at its start time, whether or not the previous rule expires before that start time.

In an ideal implementation, the new RuleList will be evaluated against any site asset that matched the reference asset associated with that rule. This can be accelerated by retaining any raw match data and then just evaluating the new rules with it rather than re-querying the recognition system. Even so, this may not be instantaneous, and it may be too expensive or complex for some UGC sites to do it at all; clarifying what can, should, or must happen for a particular combination of UGC site, recognition system, and content owner is an implementation-specific exercise.

3.4.3 Reference Asset Update

When a new reference asset is added to the system, in an ideal world all existing site assets would be tested against it immediately. Although not complex technologically – once a reference asset and a site asset are in the system, it doesn't matter which came in first – it is operationally complex because of the volume of content that has to be re-checked and consumer expectations at the UGC site if previously allowed content gets taken down.

This specification does not hinder the re-scan, although we fully expect that such a process will take time, rather than be immediate; further details of how it should happen belong in legal and operational agreements between the parties involved.

4 SPECIFICATION

This describes a way of defining detection criteria and consequences of detection in content recognition systems, which are to be used when a fingerprint is matched or when a watermark is detected. It does not define how to implement the actions. Its primary purpose is two-fold:

1. A method for communicating the desired behavior from the rights holder to another entity (e.g. a UGC site) which is expressed by the rights holder as a set of rules
2. A way of communicating a required action from a system that evaluates the rules to other systems, which consists of notifications passed from the rules system to a set of external systems.

4.1 Required External Data

This is not a completely stand-alone specification. To be implemented, it requires the following to come from some external source:

SiteAssetID – This comes from a local numbering system and is used to identify a candidate asset (e.g. an uploaded video) in the context of the running system; 'running system' could be the detection system itself or the system that is calling to it. The asset this refers to is called a 'site asset.'

OriginatorID – This identifier determines the source of the site asset and describes the origin of the content identified by SiteAssetID. It might be a transaction ID, a session ID, a user ID, or some other opaque unique identifier.

OriginalAssetID – This is how the system refers to the asset to which rules are attached, and is given as a result after the matching system has run. It could be an ID internal to the matching system, or an ISAN, or some other ID specific to a site or an ISP. The asset this refers to can be thought of as a 'master' asset, a 'reference' asset, or the 'matched' asset as well, depending on the terminology of the content identification system. It is the logical linkage between the Reference database and the Rules database.

GroupID – Some detection rules work across groups of content (e.g. all the episodes of a particular show.) Rights holders who wish to use rules about groups must, of course, create the groups and assign assets to them. Groups must be named universally uniquely. The current specification allows groups to based on UUIDs (which are highly likely to be unique, given a good UUID generator) , ISANs (which are guaranteed to be unique), and URIs. Groups are not hierarchical, but an original asset can be part of more than one group.

Each of these is referenced in the specification, and each is assumed to be available at runtime as the rules are applied and actions performed.

It is also assumed that there are external systems that can transform OriginatorIDs, SiteAssetIDs, and OriginalAssetIDs into more complex data (e.g. URLs and user details); these will be used by the system processing the triggered actions.

4.2 Rules, Criteria, and Actions

The examples given in this section are fragments; please see the use cases for complete examples.

4.2.1 RuleList Element, Asset Element, and Sub-elements

A RuleList is the top-level construct. It contains some information about itself (name, validity information, etc), information about the entity that submitted it (called 'the owner', for the sake of brevity), a list of reference assets (and information about them), and a set of rules to apply to those reference assets at content recognition time.

The list of assets is a way of simplifying having the same rules apply to multiple pieces of content. After ingestion, the behavior for a RuleList that contains an asset list $\{A_1, A_2, \ldots, A_n\}$ and a set of rules R must be the same as the system had ingested n RuleLists individually containing asset list $\{A_1\}$ and rules R, asset list $\{A_2\}$ and rules R,…, all the way through to asset list $\{A_n\}$ and rule R.

A RuleList can be ingested independently of a reference asset. After the initial ingestion of the RuleList a new version can be submitted, which replaces the existing one, subject to the constraints in section 3.4.2 above. For example, the rules may be different before and after a movie's initial DVD release. This feature can also be used to provide different rules for a single item of a set, for example changing the rules for the last episode of a series, or providing different rules for which certain rights have not been cleared or are in dispute.

RuleList Element

Element	Attribute	Definition	Value
RuleList			
	version	Current version is 1	Integer
	revision	Current revision is 1	Integer
RuleListName		Name for this set of rules; intended for incorporation into human-readable logs and statistical analysis	String
	version	Optional version number for this list of rules; no default	Integer
	revision	Optional revision number for this list of rules; no default	Integer
RuleListCreationTime		Creation time for this RuleList	XML DateTime
RuleListID		Identifier for this RuleList internal to the supplier of the list. It is intended to be	String

		something that is easier to use in automated handling of notifications and ingestion status than the RuleListName	
RuleListValidDuration		Period for which this RuleList applies. If not present, the RuleList is always valid.	See TimeInterval element
SiteConcerned		Informational field describing the site for which the rules are intended (if known.)	URI
Owner		Information about the content owner.	See Owner element
AssetList		Contains one or more assets to which the Rules apply. The file is valid if this is not present. See the Templates section.	See Asset Element
Rule		One or more individual Rule elements. If no rules are specified, no actions are taken on detection. This makes it possible to accept original assets that don't require detection rules without requiring a different ingestion path – everything has a rules file.	See Rule Element

Asset Element

Element	Attribute	Definition	Value
Asset			
OriginalAssetName		Convenience field for human readability	String
OriginalAssetID		A way of uniquely identifying a piece of content. See 'Required External Data' section.	Type-specific – see "ID Types and ID Values" table
	type	Type of the ID.	"ISAN", "UUID", "Coral", or site- or vendor-

			specific string.
AlternateURL		Used by the AlternateContent action (q.v.)	URL
AlternateInfo		Used by the AlternateContent action (q.v.)	any
Group		An asset may belong to 0 or more groups	See Group Element

4.2.1.1 ID Types and Values

ID type	**ID value**
ISAN	An <ISAN> element, as specified in ISO15706-2 Annex D.
UUID	A UUID in the form 8-4-4-4-12
URI	A URI; this allows compatibility with TVAnytime and MPEG-21
Grid	A Global Release identifier for a music video; exactly 18 alphanumeric characters
ISRC	International Standard Recording Code for music videos; exactly 12 alphanumeric characters
Coral	A Coral <Resource> element, as specified in Coral Core Architecture Specification, Version 4.0, §2.5.3
Other	An owner-specific string; this should be unique at least within the context of assets from a particular owner, and globally unique when concatenated with the OwnerDomain field of the Owner element.

IDs reasonably guaranteed to be unique, such as ISAN and UUIDs, have many advantages; types that do not guarantee uniqueness should be used with caution. The behavior of rules with non-unique asset IDs is unspecified and undefined, as is the behavior of rules for asset IDs that require external data to fully interpret.

Here are some examples of original assets that have no rules and belong to no groups. Vendor-specific and local naming schemes are easy, but may not be interoperable across sites; a UUID-based ID would be very similar, but have a better chance of working in multiple places.

```
<Asset>
  <OriginalAssetName>Bonanza Episode 1</OriginalAssetName>
  <OriginalAssetID type=”other”>0xdeadbeef</OriginalAssetID>
</Asset>
```

ISANs are well understood, and globally interoperable.

```
<Asset>
  <OriginalAssetName>The Great Dictator </OriginalAssetName>
  <OriginalAssetID type=”ISAN”>
    <ISAN root=”0000-0000-F69E”/>
  </OriginalAssetID>
</Asset>
```

URI-based naming schemes, some of which guarantee uniqueness, are quite common in the industry, and can be used with this specification.

Coral offers complete flexibility in naming schemes, through indirection in the system-identifier. (The first example above, for instance, could have a schema built for it, which could then be used as part of a Coral Resource element.) Note, though, that Coral itself does not guarantee uniqueness – that must be provided by the systems referenced inside the cor:resource tag.

```
<RuleList version="0" revision="0"
    xmlns="http://www.movielabs.com/cr/rules"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation=
        "http://www.movielabs.com/cr/rules schema/rules.xsd"
    xmlns:cor="http://www.coral-interop.org/arch/core/4-0">
 <AssetList>
  <Asset>
   <OriginalAssetName>GarbageX Demo Vid</OriginalAssetName>
   <OriginalAssetID type="Coral">
     <cor:resource primitive="true">
       <cor:system-identifier name="MMM Music">
           urn:com:indytracker
       </cor:system-identifier>
       <cor:name>
        <resource-metadata
         xmlns="http://indytracker.com/schema">
          <Artist>GarbageX</Artist>
          <Title>Combustible Rags</Title>
          <ShareCode>payg-93</ShareCode>
        </resource-metadata>
       </cor:name>
     </cor:resource>
   </OriginalAssetID>
   </Asset>
  </AssetList>
  <Owner>
    <Name>Raymond</Name>
    <Email>test@test.com</Email>
```

```
    <Email>test2@test.com</Email>
    <Phone>+440234587394</Phone>
    <Geography type="include">
      <Country>UK</Country>
      <Country>US</Country>
    </Geography>
  </Owner>
</RuleList>
```

4.2.1.2 Owner Element

This contains contact information about the owner, or entity submitting the rules. The email address or addresses will be used by some actions.

Element	Attribute	Definition	Value
Owner			
Name		Name of the owner	String
OwnerDomain		Domain associated with the owner, used as a way of uniquely identifying the owner. National subsidiaries of studios may still call themselves 'StudioName' rather than 'StudioName France' but stdioname.com and studioname.fr are unambiguous.	A domain name; informally, the second part of a URI.
Email		General administrative email address to use if others not present or appropriate	String
EmailTakeDownNotifications		Email address for notifying the owner about TakeDown	String
EmailReportToOwner		Email address for notifying the owner about ReportToOwner actions	String
EmailQuarantine		Email address for notifying about Quarantine actions	String
EmailLeaveUp		Email address for notifying about LeaveUp actions that require further action.	String
EmailConflicts		Email address to use for conflicts on rule execution	String
EmailLog		Email address for sending batched results of the Log	String

		action.	
Phone		Phone number with country code; multiple allowed	String
Geography		Describes the geographies for which the owner has exploitation rights. This can be used for certain kinds of performance optimization (knowing when rules do or don't have to be checked) and for certain kinds of ingestion-time conflict resolution (See 3.3.7.1) If not present, the owner has rights everywhere.	See CountryList element
Extra		Optional opaque information supplied by and used by the owner; it is never used by the content rules system except as data to return back to the owner.	String(4096)

Only the general email address is required. Not all content owners will want email for the rest of the items, and some UGC sites and recognition services provide information via secure ftp or RSS feeds instead. In the end, it is up to the content owner and site running the rules system to agree on a communication mechanism – these fields are present to provide a standard way of communicating some commonly requested information, not to dictate a particular workflow.

For example, a site and an owner may agree to use email for notification of TakeDown notices and Quarantine activity, but deliver ReportToOwner actions in the same file as Log actions.

See section 6.8 for more discussion of notification and reporting.

4.2.1.3 TimeInterval Element and TimecodeRange ELement

Duration is expressed as a half-open interval of the form [start, end). It can be expressed as 'after some date', 'before some date', or 'between two dates'.

Element	**Attribute**	**Definition**	**Value**
TimeInterval			

	start	Start time of the interval; if not present, it is infinitely far in the past	XML Date Time
	end	End time of the interval; if not present, it is infinitely far in the future.	XML Date Time
	duration	If this is the only element present, it is duration from now. If present with start, End is computed as start+duration If present with end, Start is computed as end – duration. If present with both, it is ignored.	XML duration

Segments within an asset are described with the TimecodeRange element.

Element	**Attribute**	**Definition**	**Value**
TimecodeRange			
SegmentStart		Start time of the range, relative to the start of the asset	A timecode of the form HH:MM:SS:FF@ff Where HH is hours, MM is minutes, SS is seconds, FF is a frame within that second, and ff is a frame rate.
SegmentEnd		End time of the range, relative to the start of the asset	A timecode of the form HH:MM:SS:FF@ff Where HH is hours, MM is minutes, SS is seconds, FF is a frame within that second, and ff is a frame rate.

4.2.1.4 Group Element

An asset may be part of zero or more groups. Each film in a movie franchise might be in the same group – e.g. there might be a group for the Hope and Crosby "Road" movies. Episodes of series might be in two groups – one for the season and one for the whole series. The only thing you can do with a group is to query whether or not a particular OriginalAssetID belongs to it; some rules are triggered based on group membership of original assets matched by a site asset.

Element	Attribute	Definition	Value
Group			Type-specific – see "ID Types and ID Values" table
	type	Domain within which the id is to be evaluated (and within which it must be unique.)	"UUID", "ISAN", "URI", or "Other"
	name	Name of the group	String

If two assets have the value and type of one of their respective Group elements in common, then those two assets are both in that group. This is true for all group types; see below for an example that uses UUIDs. The value match must be exact – no extra parsing, truncating, or interpreting is done when determining group membership in this way.

Asset IDs of type "URI" and "ISAN" generate automatic or implicit group membership. An implementation is allowed to turn the implicit group IDs into an explicit Group element and attach them to the Asset, which reduces the runtime test for group membership to string equality as described in the preceding paragraph.

An original asset with an ISAN ID is automatically a member of the group defined by the root portion of the ISAN. This means that if all members of a series use the same ISAN root, there is no need to explicitly have a Group element in the Asset description. An explicit group of type "ISAN" can still be used, for example for assets that are transitioning from other form of ID to ISAN.

For AssetIDs of type "URI", the asset is automatically a member of a implicit group named by the URI up to the last '/' character. For example an asset with and ID of type "URI" and value '//studio.com/tv/series1/episode1' is in the group '//studio.com/tv/series1,' as is the asset with ID of type "URI" and value of '//studio.com/tv/series1/episode2'. The asset with ID of type "URI" and value '//studio.com/tv/series2/episode1' is not in that group.

Here are two assets in the same group:

```
<Asset>
  <OriginalAssetName>Bonanza Episode 1</OriginalAssetName>
  <OriginalAssetID type="other">0xdeadbeef</OriginalAssetID>
  <Group type="UUID">
    5f9a3566-8df6-11dc-8314-0800200c9a66
  </Group>
</Asset>
```

```
<Asset>
  <OriginalAssetName>Bonanza Episode 2</OriginalAssetName>
  <OriginalAssetID type="other">0xbeeffeed</OriginalAssetID>
  <Group type="UUID">
    5f9a3566-8df6-11dc-8314-0800200c9a66
  </Group>
</Asset>
```

4.2.2 Rule Element and Sub-elements

Each rule in the rule list is composed of detection criteria and actions, with an optional time during which they are valid.

Rule Element

Element	Attribute	Definition	Value
Rule			
	name	Optional identifier for this rule. It is provided to any actions triggered in this rule.	String
	priority	If the detection criteria for more than one rule are met, the actions for the highest priority rule or rules get triggered.	Integer 1-100 100 is highest priority, 1 lowest
	matchedComponents	Some rules may be different based on whether the site asset matches an original in audio, video, or both. A rule is triggered only if the matched components 'match' (loosely speaking' the ones in this attribute. If not present this defaults to "any." If the recognition system cannot implement this feature, it should be treated as "any". The matched component types are made available to subsequent actions.	"audio", "video", "both", or "any"
	alwaysProcess	Always process this rule, whether or not the criteria are	Boolean

		met. The priority attribute is ignored.	
	ignoreWhiteList	If this attribute is set to true, the whitelist referenced in UGCPrinciples §3.e is to be ignored for detection and actions. This may be appropriate for some content. Implementation of the whitelist is outside the scope of this specification.	Boolean
	generateACNS	If this attribute is set to true, the system processing the action notifications must generate ACNS-format copyright infringement XML. See References for ACNS specification.	Boolean
RuleValidDuration		Time during which this rule must be applied. If not present, this rule is always applied.	TimeInterval element
IncludeSegments		Segments within the reference asset to be considered when doing the raw match. If not specified, all of the asset is implicitly included.	One or more TimecodeRange elements
ExcludeSegments		Segments within the reference asset to be ignored when doing the raw match. If not specified, none of the asset is explicitly excluded.	One or more TimecodeRange elements
DetectionCriteria		Criteria to apply when an asset is matched. When a match is detected and the criteria are met, all the actions are taken. When a match is detected and this field is not present, all the	See DetectionCriteria Element

		actions are always taken, no matter what the priority of the rule.	
Actions		Actions to take when the detection criteria are met. There must be at least one.	See Action Element

A missing DetectionCriteria element and a DetectionCriteria element with no sub-elements both count as empty sets of criteria and trigger the actions. If there are no successfully evaluated rules in the rule set with a higher priority. This can be used to provide a backstop action at the end of a series of decreasing priorities.

An implementation should:

- evaluate all rules that have AlwaysProcess true, and execute them, ignoring the priorities
- evaluate the remaining rules in priority order; a rule with no DetectionCriteria or empty DetectionCriteria with priority no lower than the current priority counts as a successful rule execution
- stop evaluating individual rules once all the rules that match the priority of the first successful rule have been evaluated

IncludeSegments and ExcludeSegments can be used for excluding commercials from broadcast TV content, for example, or ignoring film excerpts included in review shows.

If neither IncludeSegments nor ExcludeSegments is specified, the entire reference asset is considered for the rule. If only IncludeSegments are specified, only those segments of the asset are considered for the rule. If only ExcludeSegments are specified, the whole asset except for those segments is considered. If both are specified, the IncludeSegments with ExcludeSegments removed are considered.

4.2.2.1 DetectionCriteria Element

Element	Attribute	Definition	Value
DetectionCriteria			
MatchThreshold		The level of quality or certainty that must be achieved by the underlying recognition system before this rule is applied. If this field is not present, it is up to one or both of recognition system and the site to decide what counts as a good enough match. Not all recognition system will support this parameter. If a recognition system does not	

		have this information available, or is completely confident in its match/no match decisions, then it should provide '100' for anything it returns as a reasonable match.	
	percent	The level of 'sureness' that must be met for this to count as a match.	INTEGER [1-100]
MinLengthMatched		The criterion is met if the amount of matching time found between the original and the site asset is greater than or equal to the specified Time. The time required and the time found are available to subsequent actions.	
	time	The amount of time that must be exceeded for the rule to trigger	XML duration
MinAggregateLengthMatched		As with MinLengthMatched, except that the time imputed to the site asset is taken from matches to the original asset associated with the current RuleList and all assets associated with all the Groups in the RuleList. The time required, time found, and details of any matched assets found through groups are available to subsequent actions. See section 6.6 for further discussion.	
	time	The amount of time that must be exceeded for the rule to trigger	XML duration
MinPercentOfSiteAssetMatching		The criterion is met if the amount of time matched, as a percentage of the length of the local copy, is a greater than or equal to the specified Percent. The percent required and the percent found are available to subsequent actions	
	percent	The percentage of the site asset that must come from the original	INTEGER [0-100]
MinPercentOfSiteAssetAggregateMatching		As with MinPercentOfSiteAssetMatching, except that the time imputed to the local copy is taken from matches to the original asset associated with the current RuleList and all assets associated with all the Groups in the RuleList.	

		The time required, time found, and details of any matched assets found through groups are available to subsequent actions. See section 6.6 for further discussion.	
	percent	The percentage of the site asset that must come from originals.	INTEGER [0-100]
MinPercentOfOriginalAssetMatched		The criterion is met if the amount of an original that is matched by the copy is greater than or equal to Percent of the original. The percent required and the percent found are available to subsequent actions.	
	percent	The percentage of the original that must be found in the local copy.	INTEGER [0-100]
SectionMatched		The criterion is met if at least [percent] of the content appearing in the reference asset between time [start] and time [start+length] is found in the candidate asset.	
	start	An implementation is not required to support finer precision that single seconds	XML duration
	length	An implementation is not required to support finer precision that single seconds	XML duration
	percent	Minimum percent of the specified section that must match for the criterion to be met	INTEGER [1-100]
Watermark Detected		This allows for detection of some standard marks. Vendor specific marks may be included in a future version (see Appendix I)	
	type	The watermark that needs to be detected to meet the criterion. The detected mark is made available to subsequent actions. This cares only about the payload of the watermark, not its encoding; a detection system may have multiple watermark detection systems, one for each different (usually vendor-specific) way of carrying the watermark. DCI-forensic is triggered by the 35-bit DCI mark. AACS-theatrical is triggered by the presence of the AACS "Theatrical – no home use"	"DCI-Forensic", "AACS-theatrical", "AACS-consumer"

		mark. AACS-consumer is triggered by the AACS "Consumer Mark"	

Note that the criteria within an individual rule are ANDed together, and individual rules are done in order of priority.

This example triggers associated actions if more than one third of the original asset is present in the site asset:

```
<DetectionCriteria>
  <MinPercentOfOriginalAssetMatched percent="33"/>
</DetectionCriteria>
```

This example triggers associated actions if more than three quarters of the site asset comes from an original asset and the length of time matched is more than 2 minutes:

```
<DetectionCriteria>
  <MinPercentOfLocalMatched percent="75"/>
  <MinLengthMatched time = "PT2M" />
</DetectionCriteria>
```

4.2.2.2 Actions Element and sub-elements

Element	Attribute	Definition	Value
Actions		A list of actions to invoke when the detection criteria are met	Any number of simple or complex actions.

The rules and criteria are applied based on any CountryList in the Owner elements of the RuleList. Some of the actions may differ by location of the consumer; the underlying business models can be based on both 'only in' (inclusion) and 'everywhere but' (exclusion.) The actions express this using a CountryList element.

With this element, any country not explicitly included is excluded; any element not explicitly excluded is included. Often two actions will have the same country list, one with "include" and one with "exclude", for example one action to require ad support everywhere but in a particular set of countries and another to make the content unavailable for countries in that list.

Element	Attribute	Definition	Value
CountryList			

	type	Whether the list is inclusionary or exclusionary	“include” or “exclude”
Country			ISO 3166-1 alpha-2 country code

This list would disallow an action in the UK, the US, and Tuvalu:

```
<CountryList type=”exclude”>
  <Country>gb</Country>
  <Country>us</Country>
  <Country>tv</Country>
</CountryList>
```

This list would include the Baltic republics:

```
<CountryList type=”include”>
  <Country>lv</Country>
  <Country>lt</Country>
  <Country>ee</Country>
</CountryList>
```

4.2.2.2.1 Simple Actions

There are some very simple actions with simple attributes and no sub-elements, which for convenience are all grouped together in this table.

Element	Attribute	Definition	Value
NotifyOriginator		This generates a request to send some form of notification to the entity represented by OriginatorID, for example an email detailing the rules infringed by the site asset that triggered the rule.	
Log		This generates a request to save all the information about the execution of this rule somewhere, e.g. to a database or a log file. The value of the tag is saved as part of the log.	String(255)

	assertOwnership	If this is present and set to true, the content owner may be taking no action, but is asserting legal ownership of the matched content. This assertion is saved by the logging mechanism.	Boolean
ReportToOwner		This generates a request to send all the information about the execution of this rule to the persons or systems designated in the Owner element. The value of the tag is sent as well.	String(255)
	assertOwnership	If this is present and set to true, the content owner may be taking no action, but is asserting legal ownership of the matched content. This assertion is sent with the report.	Boolean

4.2.2.2.2 Complex Actions

More complex actions are described individually.

Element	Attribute	Description	Value
TakeDown		This generates a request to make the content invisible and unavailable – no external consumer should be able to find out that it was ever there in the first place. There may be legal requirements to keep the content on the system, but quarantined, for some length of time before physically deleting it. This is also an appropriate action from which to trigger a DMCA notice. TakeDown is an all or nothing activity; for more nuanced actions, please see the LeaveUp and Quarantine elements.	
	assertOwnership	If this is present and set to true, the	Boolean

		content owner is asserting legal ownership of some part of the content. Some entities (e.g. law enforcement) may want content taken down without asserting ownership. It is included in the generated Notification element and must be included in a record of TakeDown actions (see below.)	
CountryList		List of countries where the asset is or is not to be removed	CountryList element

There are consequences to TakeDown that are beyond the scope of this specification; it is hoped that industry-wide best practices for these will emerge:

- There needs to be an appeals and resolution process if the originator of the site asset contests the TakeDown action.
- A record needs to be kept of all TakeDown actions and the circumstances that led to them.

Element	**Attribute**	**Definition**	**Value**
LeaveUp		This is intended to allow initiation of a variety of activities – negotiation, expiry of legal availability windows, market tests, etc.	
	assertOwnership	Even though a decision has been made to leave the site asset available, ownership is being asserted.	Boolean
LeaveUpDuration		Length of time for this to be left up. If not present, leave up forever.	TimeInterval
CountryList		List of countries for which this action should be taken (or not)	CountryList
ExpiryActions		What to do when Duration has expired	List of Simple or complex action; the list may not include another LeaveUp action.

A reasonable action for ExpiryAction would be to quarantine the content. Sending a notification to the content owner would also be appropriate.

LeaveUp actions must be logged if they have either an ExpiryAction (so the action can occur) or assertOwnership set to true.

Element	Attribute	Definition	Value
Quarantine		This sends a request to make the content unavailable (e.g. for downloads.) It is less draconian than TakeDown, and is intended to allow for short-term legal or commercial negotiations that eventually allow the asset to be made available, or enable monitoring of how often people try to acquire this copy of the asset. Some systems may implement Quarantine and TakeDown with the same internal mechanism, but the legal and procedural follow-ons can be different.	
	assertOwnership	If this is present and set to true, the content owner is asserting legal ownership of some part of the content. It is included in the generated Notification element and must be included in a record of actions taken, similarly to TakeDown.	Boolean
CountryList		List of countries where the asset is or is not explicitly available	CountryList element

Quarantine is useful when there is uncertainty about the proper disposition of the site asset. Quarantine actions must be logged for tracking handling of the quarantine, and to provide a record of assertions of ownership.

As an example, a broadcaster might want to make its content available to residents of the UK, but not anywhere else. Note that this is done by excluding Great Britain from the countries to which not to provide the content, making it available there.

```
<Actions>
  <Quarantine>
      <CountryList type=”exclude”>
          <Country>gb</country>
      </CountryList>
  </Quarantine>
</Actions>
```

Element	Attribute	Definition	Value
AlternateContent		This uses fields from the Asset to provide alternate or additional content and information, for example for providing better quality video or an approved marketing trailer.	
	alternateInfo	If true, display the alternate information from the Asset element.	Boolean
	alternateUrlAsLink	If true, display a link to the alternate content; else, display the content itself	Boolean
	displaySiteContent	Display the SiteAsset if true	Boolean
CountryList		Countries to explicitly include or exclude	CountryList element

All the attributes default to True, meaning that the alternate info is displayed with a link to the alternate content, alongside the site asset.

Content owners and UGC sites may come to agreements about the format and display of the text in AlternateInfo and the content pointed to by AlternateURL. Standardizing and encapsulating this information is a reasonable direction for future versions of this specification.

Note that the alternate content is taken from the Asset, and so is not itself geography dependent, although the use of it is. If geography-dependent alternate content is wanted, it can be provided via geofiltering at the site providing the alternate content.

Element	Attribute	Definition	Value
OwnerAdSupported		It may be the case that infringing content is allowed to stay up if ad space is sold around it. This action sends a request to initiate that process, as a result of which ad content is retrieved. For this action, the content owner provides the ads.	
URL		Different original assets will have	URL

		different schemes for ad-supported availability, and this URL provides the necessary flexibility. See the Notification Element for details of using this URL.	
	static	If this is true, there is no requirement for the site's system to read the URL dynamically – it can be read once at content ingestion time and cached locally, for example. Otherwise, the URL needs to be read each time it is used, for example when a rotating series of ads is presented.	Boolean
CountryList		Countries to include or exclude for this action	CountryList element
AllowedType		Kinds of ads that are allowed. Multiple may be specified. Defaults to "any"	"video-pre", "video-post", "video-overlay", "banner", "text", "any"

Note: With OwnerAdSupported, the content owner provides a URL as the way of getting ads to use around the content. The content owner and the site operator will need to have negotiated an agreement about formats, types, and delivery mechanism.

Element	Attribute	Definition	Value
SiteAdSupported		It may be the case that infringing content is allowed to stay up if ad space is sold around it. This action sends a request to initiate that process. For this action, the site (perhaps via an ad service) provides the ads.	
CountryList		Countries to include or exclude for this action	CountryList element
AllowedType		Kinds of ads that are allowed. Multiple may be specified. Defaults to "any"	"video-pre", "video-post", "video-overlay", "banner", "text",

			"any"

Note: SiteAdSupported does not have a URL – it is assumed that the site's notification system manages that internally, and there is nothing the content owner can supply other than the bare bones of the action. This is a simpler mechanism than the more general AdSupported action.

Element	Attribute	Definition	Value
License		UGCPrinciples §3.c allows the content to be licensed in a way of the Copyright Holder's choosing. This action sends a request to initiate that process. The results of calling the URL can't be cached, since they will almost certainly result in a generated page specific to the asset, the originator, or the operator of the local system.	
URL		Different original assets will have different licensing schemes; this allows the originator or someone trying to access the content to be presented with a license agreement. See the Notification Element for details of using this URL.	URL
CountryList		List of countries to explicitly include or exclude	CountryList element

For a more complex example of actions, the broadcaster that only provided content to the UK in the previous example might change the model to one of available in the UK, local video ad support in the US, and unavailable elsewhere:

```
<Actions>
  <Quarantine>
      <CountryList type="exclude">
          <Country>gb</Country>
          <Country>us</Country>
      </CountryList>
  </Quarantine>
  <SiteAdSupported>
      <CountryList type="include">
            <Country>us</Country>
```

```
      </CountryList>
      <AllowedType>video-pre</AllowedType>
      <AllowedType>video-post</AllowedType>
  </SiteAdSupported>
</Actions>
```

See the Use Cases section for examples of complete RuleLists.

4.3 Notifying External Systems

A notification element contains information about actions or sets of actions and the RuleList, Rule, and DetectionCriteria that trigger them. There are several requirements for sending notifications:

If the detection criteria trigger more than one action, they are allowed to generate either one notification containing all the actions or one notification per action. It is preferred to send one notification containing all the actions.

All the criteria that trigger an action must be reported with the triggered action, and the notification must be sent only once, NOT once per criterion.

All fields for all met criteria must be set; if none of the criteria-related fields are set, the notification came from a rule with no detection criteria set.

Individual Rules operate independently of each other. After a content match and after detection criteria are met, notifications are sent for the highest priority rule or rules that have met their criteria (including rules with no DetectionCriteria at that priority), and rules with AlwaysProcess set to true. This may result in the sending of multiple Notification elements.

Individual RuleLists operate independently of each other. This means that if a site asset triggers actions from two separate original assets – e.g. Star Wars and Saturday Night Live – then the Notification elements from both original assets are sent. This also means that each element of a group that ends up triggering actions in an aggregate matching criterion will send Notification elements -- e.g. if three sketches from three episodes of The Muppet Show are detected and their total time exceeds the time in an AggregatedMatch rule three Notification elements will be generated. Put another way, a Notification only ever has a single Asset and a single SiteAsset

The requirements for acting on notifications are:

Actions in notification elements that have URLs must do an http POST to the URL with post data of notification=NotificationElement and, if the GenerateACNS element is present, additional post data of ACNS=<generated ACNS information> . See References for a specification of the information required.

A record should be kept of actions taken with infringing content, as outlined in UGCPrinciples §10b. The Notification Element is intended to contain sufficient information for this record, in addition to its role as a communication mechanism. Further details on this can be found in section 6.8.

4.3.1 Notification Element

Element	Attribute	Definition	Value
Notification			
	version	Taken from original rules	
	revision	Taken from original rules	
	ignoreWhiteList	Taken from the rule that triggered this action. If the attribute is present, the whitelist referenced in UGCPrinciples §3.e is ignored.	Boolean
	generateACNS	Taken from the rule that triggered this action. If the attribute is present, ACNS data gets sent as POST data to any external URLs (see above.)	Boolean
RuleListName		Taken from original rules	String
	version	Taken from original rules	Integer
	revision	Taken from original rules	Integer
RuleListCreationTime		Taken from original rules	XML DateTime
RuleListID		Taken from original rules	String
Owner		From the original rules	See Owner element
Asset		Taken from the RuleList whose evaluation triggered the action	See Asset element
RuleName		Name of the particular rule that generated this notification	String
	Priority	Taken from the rule; set to 100 for rules with no detection criteria	Integer 1-100
RuleListValidDuration		Taken from the RuleList	TimeIntervl element
RuleValidDuration		Taken from the rule	TimeInterval element
IncludeSegments		Taken from the rule	One or more TimecodeRange elements
ExcludeSegments		Taken from the rule	One or more TimecodeRange elements
SiteConcerned		Taken from the RuleList	URI
SiteAsset		Information about the site asset that matched the original	See SiteAsset element
MatchedComponents		The components that were matched to trigger the criteria. A system that	“audio”, “video”, “both”,

		does not know how to differentiate components should return “any”; a system that identifies only one kind of component should return that component.	or “any”
OriginatorID		How the system uniquely identifies the entity that created the site asset	String
	Country	The location of the originator, if known. If not present, assumed to be unknown.	ISO 3166 country code
Actions		Actions to be performed. See notes above on single vs. multiple.	See Action Element
MatchThreshold		Present if matchThreshold criterion was met	
	required	From the original rule	INTEGER [1-100]
	observed	Not all recognition system will support this level of granularity. If a recognition system does not have this information available, it should provide ‘100’ for anything it returns as a reasonable match.	INTEGER [1-100]
LengthMatched		Present if a length criterion was met	
	required	From the original rule	XML duration
	matched	Length of time matching between local and original	XML duration
AggregateLengthMatched		Present if AggregatedLengthMatched was met	
	required	From the original rule	XML duration
	totalMatched	Total amount of the site asset matched; This should be the same as LengthDetected attribute of SiteAsset.	XML duration
	matchedFromThisOriginal	The amount found using the original asset to which this notification applies	XML duration
PercentOfLocalMatched		Present if PercentOfLocalMatching was met	
	required	From the original rule	Integer [0-100]
	matched	Percent of local copy found in the original	Integer [0-100]
PercentOfOriginalMatched		Present if	

		PercentOfOriginalMatched was met	
	required	From the original rule	Integer [0-100]
	matched	Percent of the original found in the local copy	Integer [0-100]
AggregatedPercent LocalMatched		Present if AggregatedPercentLocalMatched was met	
	required	From the original rule	Integer [0-100]
	totalMatched	The total amount of the site asset that was matched; This should be the same as LengthDetected attribute of SiteAsset.	XML duration
	percentFromThis Original	Amount of the site asset matched by this source	Integer [0-100]
	timeFromThisOri ginal	Time in the site asset detected from this source	XML duration
SectionMatched		Present if SectionMatched was met	
	start	From the original rule	XML duration
	length	From the original rule	XML duration
	percentRequired	From the original rule	INTEGER [1-100]
	percentMatched	Percent of the interval [start, start+length] from the reference asset found in the matched asset	INTEGER [1-100]
WatermarkDetected		Present if WatermarkDetected was met	See WatermarkDetected element

4.3.2 WatermarkDetected Element

Element	**Attribute**	**Description**	**Value**
WatermarkDetected		The value of the watermark, in a canonical form	String
	type	The type of mark	"DCI-forensic", "AACS-theatrical", or "AACS-consumer"

4.3.3 SiteAsset Element

Element	Attribute	Definition	Value
SiteAsset		Everything that is known about a site asset	
SiteAssetID		How this asset is identified locally. It may be a URL, or an ID that is opaque outside of the context of the site, or anything else.	String
SiteDomain		This is the site at which the SiteAsset was found (if known.) Some content owners may want different treatment for different sites; this field provides information to allow them to do that, as well as enabling certain kinds of tracking and statistics.	URL
TimeCreated		Time the site asset was created, if known. It could, for example, be the time it was uploaded.	XML date/time
TimeMatchRequested		Time it was decided to try to match the item in question.	XML date/time
TimeMatchDetected		Time the match was detected.	XML date/time
Format		The format of the site asset, reported for statistical purposes.	String, based on 'type' attribute
	type	"FileExtension" or "MIME"	
Length		Length of the site asset	XML duration
LengthDetected		Length of identified content in this asset. For aggregated matches, this is the total of all identified segments.	XML duration

The Time sub-elements on the SiteAsset Element are:

- TimeCreated -- the creation time of the site asset (which may not be known),
- TimeMatchRequested – the time at which the system decided that the asset needed to be scanned
- TimeMatchDetected -- the time identification process completed

A system that checks all uploaded video at upload time would have the first two fields very close to equal; the results for a batch process, such as described in UGCPrinciples §3.h,

might differ substantially. These three fields can be used to determine expeditiousness of detection processes.

See the Use Cases section for examples of Notifications.

4.4 Templates

The items described here enable the separate distribution of rule templates and lists of assets. This is useful, for instance, if a standard set of rules is pre-defined for all the episodes of a TV show, even those which have not yet been release. The template can be defined ahead of time and each episode can refer to that template as the episode becomes available. This mechanism also allows the rules themselves to be treated as assets and included in business workflow as objects or components with their own identifiers.

These items are defined separately here to make it easier to explain how they work. They affect which rules get associated with an asset, not the execution of the rules, and so are somewhat different from the other attributes and elements.

When a RuleList with a templateID is ingested, it is validated and saved in the database. If a RuleList with the same templateID already exists, it is replaced.

When an AssetsWithTemplate is ingested, the existence of the template is checked; if the template does not exist, an error is returned. If it does exist, each Asset in the included asset list is connected by reference to the template. If the asset already has a rule attached, either by reference or by instance, the reference to the template replaces it. When the asset is recognized, the template rule is looked up and used.

When a RuleList that has a template ID and an AssetList sub-element is ingested, the rule list is installed as a template and the included assets refer to that template by reference. The resulting system behavior is the same as ingesting the rule list with the template ID, and then ingesting an AssetsWithTemplate referring to that template.

When a RuleList without a template ID is ingested, all assets in the included asset list receive their own copy of the rule list. If any of the assets already have a rule attached (whether by reference to a template or as an instance), the new instance of the new rule replaces it. The rule used is the instance of the rule associated with the asset.

4.4.1 RuleList

There is one new optional attribute for a RuleList

Element	Attribute	Definition	Value
RuleList			
	templateID	Identifier to use when referring to this RuleList as a template.	UUID

4.4.2 AssetsWithTemplate

An AssetsWithTemplate is sent as a top-level entity. The assets in it retrieve rules from the template by reference.

Element	Attribute	Definition	Value
AssetsWithTemplate			
	version	Current version is 1	Integer
	revision	Current revision is 1	Integer
AssetListID		Optional identifier; currently used only when returning status at ingestion time	String
SiteConcerned		Informational field describing the site for which the rules are intended (if known.)	URI
TemplateID		ID of the rule template to use for the assets in this AssetList. Exactly one such sub-element must be present in an AssetsWithTemplate element.	UUID
Owner			See Owner element
AssetList		Contains one or more Asset elements	See Asset element

4.4.3 Notification

Element	Attribute	Definition	Value
Notification			
TemplateID		ID of the template used for the rule that generated this notification	UUID

4.5 Ingestion Status

Not all RuleLists will be ingested correctly – some will succumb to transmission errors, and some will have incorrect XML or conflicts with other rules. Here is the minimum set of status codes that should be supported:

- Parsed – the RuleSet was valid XML, and parsed meaningfully. Since this is so, this status and its sub-status can return the Owner, RuleListName, and RuleListCreationTime from the RuleList
    - Sub-status: success – the RuleList parsed correctly and the rules have been added to the system for the assets.

- Sub-status: conflict – Even though the rules were parsed, there were unresolvable conflicts with other rules in the system. Notification about this should be sent to the NotifyConflict email address form the Owner element. Ideally, information about the conflicting assets will be returned as well. This does NOT count as successful ingestion, so no rules should be installed for any assets in the AssetList

- NotParsed – Some error happened, and the rules could not be parsed. This may not be able to return any information along with the status, but if anything is available (e.g. identifying information from the RuleList) it should be returned.
- MissingTemplate – If an AssetList is provided either with no template or a reference to a non-existent template, this error should be returned

5 CLASSES OF IMPLEMENTATION

The baseline is intended to be implementable on all current content recognition systems, with minimal extra work from a UGC site. The full version exploits some features that may not be available yet in all recognition systems, and requires more complex business operations infrastructure at the UGC site.

The templating features are required fro both baseline and full implementations.

5.1 Baseline Implementations

The baseline is a strict subset of the full specification. Note that the required Actions can be grouped into three classes -- upload management, presentation management, and reporting – and items within each class might be expected to share some infrastructure.

The baseline includes the following elements (with all their sub-elements and attributes, unless otherwise mentioned) :

- The full RuleList element
- The full Asset element except:
  - Group sub-element
- The full Owner element
- The TimeInterval Element, used only as a sub-element of RuleList
- The full CountryList element
- All ID types except:
  - Coral
- The full Rule element except:
  - RuleValidDuration sub-element
  - IncludeSegments and ExcludeSegments sub-elements
- The following DetectionCriteria:
  - MinLengthMatched, and all its fields
  - MinPercentOfSiteAssetMatching and all its fields
  - MinPercentOfOriginalAssetMatched and all its fields
- The following Action elements (class as defined in 5.1 in parentheses) and all their fields:
  - TakeDown (upload management)
  - Quarantine (upload management)
  - SiteAdSupported, except that AllowedType is allowed to be “any” (presentation management)

- AlternateContent , except that only the default values of the attributes are required to be supported (presentation management)
- NotifyOriginator (reporting)
- ReportToOwner (reporting)
- Log (reporting)

- The Notification element, with the following exceptions:
  - The RuleValidDuration field
  - The IncludeSegments and ExcludeSegments sub-elements
  - DetectionCriteria other than MinLengthMatched, MinPercentOfSiteAssetMatching, and MinPercentOfOriginalAssetMatched
  - Actions other than TakeDown, Quarantine, SiteAdSupported, AlternateContent, NotifyOriginator, ReportToOwner, Log

It does not include:

- The Group element
- TimeInterval on anything except the RuleList itself
- TimecodeRange element
- Any other DetectionCriteria
- Any other Actions
- The WatermarkDetected element

5.2 Full Implementations

With the following exceptions, all features in this specification are required for a "full" implementation:

- OwnerAdSupported – If this is not supported, a system is allowed to reject rule sets containing that action at ingestion time.
- MatchQuality – If a system does not support this, it should treat match quality as binary, and return either 0 or 100 when this information is needed.

`

5.3 Defaults

Most items required for operating a system based on this specification are explicitly required. The exceptions are:

- A missing country list means the item in question applies to all geographies.
- A missing duration means the item in questions applies forever (subject to the hierarchy expiration times discussed in §3.4.1.

There is no default Action; every RuleList must have at least one action in it.

6 USE CASES AND BEST PRACTICES

From these examples, it should be clear that the rules are ideally generated by an automated tool or from templates, rather than manually.

6.1 Percent of original matching; use of 'priority' attribute

A content owner might wish to have different rules based on the amount of an original asset found in the site asset. This can be done using the template:

```
<Rule priority=N>
   <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched = M>
`     </DetcetionCriteria>
</Rule>
```

With lower values of N and M for each lesser gradation required. (Other types of criteria can be used as well or instead, of course.)

As a good example of using missing DetectionCriteria as a defaulting mechanism, if the last rule in such a sequence has no DetectionCriteria element it will always be executed if no higher priority rule has applied.

This example looks at potential infringement based on the percentage of an original asset contained in a site asset. The RuleList will:

- Take the content down if it contains more than 25% of the original, generate an ACNS notice, and notify the owner
- Leave the content up and insert an ad if it contains between 5% and 25% of the original, and tell the owner for tracking purposes.
- Notify the owner if it contains less than 5% of the original; this can be used for tracking incipient popularity, general public consciousness and use of the asset, etc.

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
   xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
   xmlns="http://www.movielabs.com/cr/rules"
   xmlns:isan="http://www.isan.org/ISAN/isan"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Owner>
    <Name>MovieLabs</Name>
    <OwnerDomain>www.movielabs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
    <Geography type="include">
      <Country>us</Country>
    </Geography>
  </Owner>
   <AssetList>
     <Asset>
```

```
        <OriginalAssetName>Modern Times</OriginalAssetName>
        <OriginalAssetID type="ISAN">
          <isan:ISAN root="0000-0000-48E3" />
        </OriginalAssetID>
      </Asset>
  </AssetList>
  <Rule name="TooMuch" generateACNS="true" priority="100">
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="25" />
    </DetectionCriteria>
    <Actions>
      <TakeDown assertOwnership="true"/>
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="RevenuePotential" priority="50">
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="5" />
    </DetectionCriteria>
    <Actions>
      <ReportToOwner />
      <SiteAdSupported>
        <AllowedType>video-pre</AllowedType>
        <AllowedType>video-post</AllowedType>
      </SiteAdSupported>
    </Actions>
  </Rule>
  <Rule name="BuzzTracker" priority="10">
    <Actions>
      <Log />
    </Actions>
  </Rule>
</RuleList>
```

The notifications generated by an uploaded asset containing almost all of the original video overdubbed with a new soundtrack might be:

```
<?xml version="1.0" encoding="utf-8"?>

<Notification revision="1" version="1"
 xsi:schemaLocation="http://www.movielabs.com/cr/notification
notification.xsd"
 xmlns="http://www.movielabs.com/cr/notification"
 xmlns:isan="http://www.isan.org/ISAN/isan"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Asset>
    <OriginalAssetName>Modern Times</OriginalAssetName>
    <OriginalAssetID type="ISAN">
      <isan:ISAN root="0000-0000-48E3" />
    </OriginalAssetID>
  </Asset>
  <Owner>
    <Name>MovieLabs</Name>
```

```
            <OwnerDomain>www.movielabs.com</OwnerDomain>
      <Email>info@movielabls.com</Email>
      <Phone>+440234587394</Phone>
      <Geography type="include">
         <Country>us</Country>
      </Geography>
   </Owner>

   <RuleName priority="100">TooMuchTaken</RuleName>
   <SiteAsset>
      <SiteAssetID>usr/noname/cooltv.wmv</SiteAssetID>
      <SiteDomain>www.ugc-r-us.com</SiteDomain>
      <TimeMatchRequested>2007-11-27T12:45:00</TimeMatchRequested>
      <TimeMatchDetected>2007-11-27T14:45:00</TimeMatchDetected>
      <Format type="FileExtension">wmv</Format>
      <Length>PT1004199059S</Length>
      <LengthDetected>PT1004199059S</LengthDetected>
   </SiteAsset>
   <MatchedComponents>video</MatchedComponents>
   <OriginatorID country="us">customer90210</OriginatorID>
   <Actions>
      <TakeDown assertOwnership="true" />
      <NotifyOriginator />
      <ReportToOwner />
   </Actions>
   <PercentOfOriginalMatched required="45" matched="98" />
</Notification>
```

6.2 Absolute amount matching; country-dependent actions

As discussed above, sometimes different actions will be defined for different countries. Conceptually, if countries A, B, and C have individual actions, and the rest of the world has another action, the action list is modeled on:

<Actions>
   <Action for rest of world>
      <CountryList type=”exclude”>
         <Country>A</Country>
         <Country>B</Country>
         <Country>C</Country>
      </CountryList>
   </Action for rest of world>
   Then, for X in {A, B, C}
   <Action for country X>
      <CountryList type=”include”>
         <Country>X</Country>
      </CountryList>
   </Action for country X>
</Actions>

This example looks at potential infringement based on an absolute threshold amount of original content found in the site asset. The rules are:

- If more than 5 minutes found
  - Content is freely available in the UK
  - Content is site ad-supported in the US
  - Content is replaced with a URL from the owner everywhere else, to publicize legitimate sources
- If less than 5 minutes found
  - Content is available in UK and US
  - Content is owner ad-supported elsewhere, to generate revenue or to publicize legitimate sources

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <Owner>
      <Name>MovieLabs</Name>
                 <OwnerDomain>www.movielabs.com</OwnerDomain>
      <Email>info@movielabls.com</Email>
      <Phone>+440234587394</Phone>
   </Owner>

<AssetList>
       <Asset>
          <OriginalAssetName>Dr Who Visits White City</OriginalAssetName>
          <OriginalAssetID type="ISAN">
             <isan:ISAN root="0BBB-0BBB-0CCC" />
          </OriginalAssetID>
          <AlternateURL>
              http://promo.gallifrey.com/acquire-legit-version
          </AlternateURL>
          <AlternateInfo>
              Take a look at the official archives on Gallifrey!
          </AlternateInfo>
       </Asset>
    </AssetList>
   <Rule name="Lots" generateACNS="true"
         priority="100" matchedComponents="both">
      <DetectionCriteria>
         <MinLengthMatched time="PT5M" />
      </DetectionCriteria>
      <Actions>
         <AlternateContent>
           <CountryList type="exclude">
             <Country>gb</Country>
             <Country>us</Country>
```

```
          </CountryList>
        </AlternateContent>
        <SiteAdSupported>
          <CountryList type="include">
            <Country>us</Country>
          </CountryList>
          <AllowedType>video-pre</AllowedType>
          <AllowedType>video-post</AllowedType>
        </SiteAdSupported>
        <LeaveUp>
          <CountryList type="include">
            <Country>uk</Country>
          </CountryList>
        </LeaveUp>
      </Actions>
    </Rule>
    <Rule name="Some" priority="50">
      <DetectionCriteria>
        <MinLengthMatched time="PT1M" />
      </DetectionCriteria>
      <Actions>
        <OwnerAdSupported>
          <URL static="false">
             http://promo.aab.com/advertise-legit-version
           </URL>
          <CountryList type="exclude">
            <Country>gb</Country>
            <Country>us</Country>
          </CountryList>
        </OwnerAdSupported>
        <LeaveUp>
          <CountryList type="include">
            <Country>gb</Country>
            <Country>us</Country>
          </CountryList>
        </LeaveUp>
      </Actions>
    </Rule>
</RuleList>
```

6.3 Validity periods; expiry of LeaveUp action

It might be known ahead of time that a set of rules will change, or it might not. If a new set of rules is needed to replace a existing one, it is essential that the new version be distributed with a validity period in the rule set, to make changeover uniform and global.

If it is known ahead of time that a set of rules will change at some time in the future, there are two ways of doing it:

- Distribute the new set separately from the original set. This has the advantage of having two simpler sets of rules rather than one composite set. It has the disadvantage that it might require a second distribution to the recipients (unless their infrastructure can accept two rule sets at once.)

- Distribute one set of rules that has some rules valid before the changeover date, and some valid after. This has the disadvantage of complexity in the rules and the advantage of a single distribution. A further advantage is that it is easier to manage operational and technical aspects of the LeaveUp action.

In this use case, the rules define:

- Before date 1, UGC is used, within reason (based on percentage of an original detected), as extra-budgetary marketing material
  - If not too much is present
    - Content is available up to a certain date, after which it is to be taken down.
  - Otherwise it is taken down immediately.
- Between dates 1 and 2
  - The content is available in one country to deal with issues of free catch-up through other means; after that, it is ad-supported through the site.
  - It is unavailable elsewhere.
- After Date 2, a different release window is enforced, for ad –supported catch-up
  - The content is available with ad support through the site in one country.
  - It is unavailable elsewhere (though later the owner may change this to a promotional replacement, for instance.)

In all cases, site assets with minimal content from the original asset are ignored.

The structure of these rules is:

- Two rules in the pre-release window, neither geography-dependent
- One rule in the free catch-up window, with two geographically distinct actions
- One rule in the window after free catch-up, with two geographically distinct actions

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Owner>
    <Name>MovieLabs</Name>
                <OwnerDomain>www.movielabs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
  </Owner>
  <AssetList>
    <Asset>
      <OriginalAssetName>The NY-LA Shuffle</OriginalAssetName>
      <OriginalAssetID type="ISAN">
        <isan:ISAN root="0a0a-0b0b-0c0c" />
       </OriginalAssetID>
     </Asset>
  </AssetList>
  <Rule name="Pre-release Not OK" generateACNS="true" priority="100">
```

```
    <RuleValidDuration end="2007-12-25T00:00:00" />
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="25" />
    </DetectionCriteria>
    <Actions>
      <TakeDown />
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="Pre-release temporary OK" generateACNS="true" priority="75">
    <RuleValidDuration end="2007-12-25T00:00:00" />
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="2" />
    </DetectionCriteria>
    <Actions>
      <LeaveUp>
        <LeaveUpDuration end="2007-12-25T00:00:00" />
        <ExpiryActions>
          <TakeDown />
          <NotifyOriginator />
        </ExpiryActions>
      </LeaveUp>
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="free catch-up period" priority="75">
    <RuleValidDuration start="2007-12-25T00:00:00"
                       end="2008-01-24T00:00:00" />
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="2" />
    </DetectionCriteria>
    <Actions>
      <ReportToOwner />
      <LeaveUp>
        <CountryList type="include">
          <Country>us</Country>
        </CountryList>
        <LeaveUpDuration start="2007-12-25T00:00:00"
                         end="2008-01-24T00:00:00" />
        <ExpiryActions>
          <SiteAdSupported>
            <CountryList type="include">
              <Country>us</Country>
            </CountryList>
            <AllowedType>video-pre</AllowedType>
            <AllowedType>video-post</AllowedType>
          </SiteAdSupported>
        </ExpiryActions>
      </LeaveUp>
      <Quarantine>
        <CountryList type="exclude">
          <Country>us</Country>
        </CountryList>
      </Quarantine>
```

```
    </Actions>
  </Rule>
  <Rule name="free period over" priority="100">
    <RuleValidDuration start="2008-12-24T00:00:00+01:00" />
    <DetectionCriteria>
      <MinPercentOfOriginalAssetMatched percent="2" />
    </DetectionCriteria>
    <Actions>
      <SiteAdSupported>
        <CountryList type="include">
          <Country>us</Country>
        </CountryList>
        <AllowedType>video-pre</AllowedType>
        <AllowedType>video-post</AllowedType>
      </SiteAdSupported>
      <Quarantine>
        <CountryList type="exclude">
          <Country>us</Country>
        </CountryList>
      </Quarantine>
    </Actions>
  </Rule>
</RuleList>
```

6.4 Percent of site asset matching; matching individual components

Permitted use decisions can be based on the percent of the site asset that comes from an original asset; this can be useful when thinking about the volume of commentary, criticism, or transformation relative to the amount of original work present. This example is based on rules found in *UGC Fair Use*:

- If more than 90% of as site asset matches an original in both audio and video it is taken down, an ACNS notice is sent, and the content owner and the originator are notified.
- If a site asset matches more than 90% of an original in audio or video (but not both) the originator and owner are notified. This example does not assert ownership in this case, but it is possible to do so.

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Owner>
    <Name>MovieLabs</Name>
              <OwnerDomain>www.movielbs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
    <Geography type="include">
      <Country>us</Country>
    </Geography>
```

```
  </Owner>
  <AssetList>
    <Asset>
      <OriginalAssetName>My Way</OriginalAssetName>
      <OriginalAssetID type="ISAN">
        <isan:ISAN root="effe-ffef-feff" />
      </OriginalAssetID>
    </Asset>
  </AssetList>
  <Rule name="TooMuch" generateACNS="true"
        matchedComponents="both" priority="100">
    <DetectionCriteria>
      <MinPercentOfSiteAssetMatching percent="90" />
    </DetectionCriteria>
    <Actions>
      <TakeDown assertOwnership="true"/>
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="MarginalAudio" matchedComponents="audio" priority="50">
    <DetectionCriteria>
      <MinPercentOfSiteAssetMatching percent="90" />
    </DetectionCriteria>
    <Actions>
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="MarginalVideo" matchedComponents="video" priority="50">
    <DetectionCriteria>
      <MinPercentOfSiteAssetMatching percent="90" />
    </DetectionCriteria>
    <Actions>
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
</RuleList>
```

6.5 Multiple detection criteria

Multiple detection criteria in a rule must all be met for the actions to be triggered. This can be used to add extra flexibility. For example, here is a rule set that expresses:

- If the site asset is more than 33% taken from the original asset, and it contains more than 2 minutes from the original asset, then take it down.

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
  <Owner>
    <Name>MovieLabs</Name>
                <OwnerDomain>www.movielabs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
  </Owner>
  <AssetList>
    <Asset>
      <OriginalAssetName>The Day of the Jackal</OriginalAssetName>
      <OriginalAssetID type="ISAN">
        <isan:ISAN root="0000-0000-1CAD" />
      </OriginalAssetID>
    </Asset>
    <Asset>
      <OriginalAssetName>Three Days of the Condor</OriginalAssetName>
      <OriginalAssetID type="ISAN">
        <isan:ISAN root="0000-0001-3612" />
      </OriginalAssetID>
    </Asset>
  </AssetList>
  <Rule name="TooMuch" generateACNS="true" priority="100">
    <DetectionCriteria>
      <MinPercentOfSiteAssetMatching percent="33" />
      <MinLengthMatched time="PT2M" />
    </DetectionCriteria>
    <Actions>
      <TakeDown />
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
</RuleList>
```

This allows a site asset that embeds a small amount of an original for purposes of review or illustration, but disallows content that is just a capture of part of an asset – 1 minute of video embedded in a 3 minute review or video essay would fine, as would including 2 minutes in the background images of a substantially longer work. A stand-alone 2 minutes from the end of a thriller would not be allowed. In a real-world situation this rule would probably be deployed as one of a set of rules, rather than on its own.

6.6 Groups and aggregate matching

Sometimes assets are viewed as components of a larger entity, such as episodes of a TV series or elements of an umbrella brand; content owners may care as much about the total amount of a series in a site asset as they do about the amount of any individual item from that series, and individual assets can be part of more than one group. The strength of this kind of connection can be seen in the prevalence of mashups on UGC sites – favorite or most important scenes from multiple episodes of a TV sitcom, strung-together footage from multiple installments of an action-adventure franchise, etc.

MinAggregateLengthMatched and MinPercentOfSiteAssetAggregateMatching allow rules that deal with groups. This example:

- Takes the content down if it has more than 2 minutes of the original asset in it
- Takes the content down if it has more than 4 minutes of content in it from this asset or any other asset in its group

In this example, the AssetList in the RuleList contains two episodes of the series, though each one could just as well be in a separate RuleList.

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Owner>
    <Name>MovieLabs</Name>
                <OwnerDomain>www.movielabs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
    <Geography type="include">
      <Country>us</Country>
    </Geography>
  </Owner>
   <AssetList>
     <Asset>
        <OriginalAssetName>Leave It To Beaver 1</OriginalAssetName>
        <OriginalAssetID type="ISAN">
          <isan:ISAN root="0000-0000-80CD" episodeOrPart="0001" />
        </OriginalAssetID>
        <Group type="ISAN">
           <isan:ISAN root="0000-0000-80CD" />
        </Group>
     </Asset>
     <Asset>
        <OriginalAssetName>Leave It To Beaver 2</OriginalAssetName>
        <OriginalAssetID type="ISAN">
          <isan:ISAN root="0000-0000-80CD" episodeOrPart="0002" />
        </OriginalAssetID>
        <Group type="ISAN">
           <isan:ISAN root="0000-0000-80CD" />
        </Group>
     </Asset>
  </AssetList>
  <Rule name="TooMuch" generateACNS="true" priority="100">
    <DetectionCriteria>
      <MinLengthMatched time="PT2M" />
    </DetectionCriteria>
    <Actions>
      <TakeDown />
```

```
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
  <Rule name="TooMuchAggregate" generateACNS="true" priority="90">
    <DetectionCriteria>
      <MinAggregateLengthMatched time="PT4M" />
    </DetectionCriteria>
    <Actions>
      <TakeDown />
      <NotifyOriginator />
      <ReportToOwner />
    </Actions>
  </Rule>
</RuleList>
```

If the first rule is triggered, there will be one notification, using the single asset that triggered it. If the second rule is triggered, there should be multiple notifications, one for each asset in the group that contributed to the aggregated matching, although that may be depend on the recognition vendor's optimization strategy.

6.7 Templates

This example contains a template that:

- Takes the content down if more than 3 minutes is found matching both audio and video
- Quarantines the content if more than 3 minutes is found matching just the video

```
<?xml version="1.0" encoding="utf-8"?>

<RuleList templateID="f8a0afe0-41fb-11dd-ae16-0800200c9a66"

 revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Owner>
    <Name>MovieLabs</Name>
    <OwnerDomain>www.movielabs.com</OwnerDomain>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
  </Owner>
  <Rule name="TooMuch" generateACNS="true" matchedComponents="both"
priority="100">
    <DetectionCriteria>
      <MinLengthMatched time="PT3M" />
    </DetectionCriteria>
    <Actions>
      <TakeDown assertOwnership="true" />
      <NotifyOriginator />
      <ReportToOwner />
```

```
    </Actions>
  </Rule>
  <Rule name="MarginalVideo" matchedComponents="video" priority="80">
    <DetectionCriteria>
      <MinLengthMatched time="PT3M" />
    </DetectionCriteria>
    <Actions>
      <NotifyOriginator />
      <ReportToOwner />
      <Quarantine assertOwnership="true" />
    </Actions>
  </Rule>
</RuleList>
```

This AssetsWithTemplate element applies he template to two assets:

```
<?xml version="1.0" encoding="utf-8"?>
<AssetsWithTemplate revision="1" version="1"
xsi:schemaLocation="http://www.movielabs.com/cr/rules rules.xsd"
xmlns="http://www.movielabs.com/cr/rules"
xmlns:isan="http://www.isan.org/ISAN/isan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <TemplateID>f8a0afe0-41fb-11dd-ae16-0800200c9a66</TemplateID>
   <Owner>
      <Name>MovieLabs</Name>
                 <OwnerDomain>www.movielabs.com</OwnerDomain>
      <Email>info@movielabls.com</Email>
      <Phone>+440234587394</Phone>
   </Owner>
     <AssetList>
       <Asset>
          <OriginalAssetName>Torchwood ep 1</OriginalAssetName>
          <OriginalAssetID type="ISAN">
            <isan:ISAN root="0000-0001-CE6F" episodeOrPart="0001" />
          </OriginalAssetID>
          <Group type="ISAN">
             <isan:ISAN root="0000-0000-80CD" />
          </Group>
       </Asset>
       <Asset>
          <OriginalAssetName>Torchwood ep 2</OriginalAssetName>
          <OriginalAssetID type="ISAN">
            <isan:ISAN root="0000-0001-CE6F" episodeOrPart="0002" />
          </OriginalAssetID>
          <Group type="ISAN">
             <isan:ISAN root="0000-0000-80CD" />
          </Group>
       </Asset>
   </AssetList>
</AssetsWithTemplate>
```

6.8 Managing Trailers

Content owners sometimes want to allow, or even encourage, proliferation of certain assets; this is especially true of trailers and other promotional content.

The easiest way to make sure such content is allowed is to always upload it from an account included on the UGC site's whitelist. This means that the initial upload will not go through any recognition checking.

Allowing such content to be uploaded from non-authorized sources is trickier. The current specification does allow an ideal solution, but it is possible to permit full uploads of trailers from non-authorized sources.

To allow uploaders to make copies of the trailer and distribute them, create a rule set for the trailer reference asset that allows content where most of the trailer is in the candidate asset, and most of the candidate asset is found in the trailer. In terms of the rules in this specification, that means that MinPercentOfSiteAssetMatching and MinPercentOfOriginalAssetMatched are both close to 100%. 'Close' will depend on the details of the recognition system.

This works for people sending the trailer around, but not for embedding it in other content. The more general case of allowing the trailer to be embedded works better if the trailer contains some amount of content that is not in the parent asset (the one the trailer is for.) In that case, the owner must:

- Create a rule set with a rule that is higher priority than any rule that is associated with the underlying asset or any of its derivative works.
- Allow content that matches at least (100-N)% of the reference asset (where N is the percentage of content in the trailer that is not in the parent asset.) This ensures that some of the trailer material is in the candidate asset, which would not be the case for a candidate asset that had only the parts of the trailer than were not unique to the trailer.

This would be expressed as:

```
<?xml version="1.0" encoding="utf-8"?>
<RuleList version="1" revision="0" xmlns="http://www.movielabs.com/cr/rules"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.movielabs.com/cr/rules schema/rules.xsd"
   xmlns:isan="http://www.isan.org/ISAN/isan">
  <AssetList>
    <Asset>
      <OriginalAssetName>Blockbuster Trailer1</OriginalAssetName>
      <OriginalAssetID type="ISAN">
        <isan:ISAN root="0123-0456-0789" />
      </OriginalAssetID>
    </Asset>
  </AssetList>
  <Owner>
    <Name>MovieLabs</Name>
    <Email>info@movielabls.com</Email>
    <Phone>+440234587394</Phone>
```

```
     <Geography type="include">
        <Country>us</Country>
     </Geography>
   </Owner>
    <Rule name="AllThere" priority = "100">
        <DetectionCriteria>
            <MinPercentOfOriginalAssetMatched percent="93"/>
        </DetectionCriteria>
        <Actions>
            <LeaveUp/>
            <ReportToOwner/>
        </Actions>
    </Rule>
    <Rule name="CouldBeTrouble" priority="25">
        <Actions>
            <Quarantine/>
            <ReportToOwner/>
        </Actions>
    </Rule>
</RuleList>
```

This rule assumes that the trailer has 10% extra material in it. If some of that extra material (with 3% extra to deal with imprecision in editing and detection systems) as well as all of the original content is present, then the candidate asset is probably the trailer, or something that includes the trailer, and so is allowed. If less than that matches, the content is unavailable until the rights holder determines what to do; if the length of the trailer is below the threshold for authorized use from the parent asset, then things are much simplified.

Another way to do this would be to require matching of individual segments of the trailer using MatchThreshold, but not all detection systems have fine enough precision to deal with the short segments that make up trailers.

Yet another way would be to use a watermark-based system for generating and detecting trailers.

6.9 Communication, Log files, and Email

The specification supports the collection of information about content uploading from the recognition system and the rules engine. It does not cover the collection of information about content viewing, although this has been suggested as a future enhancement (See Appendix II.)

The following table covers two things:

- The intent behind the informational actions, and actions that generate information as a side-effect
- Some ways of distributing the information once it has been collected or generated -- the document does not formally specify this, but enough input has been received that some common practices can be sketched out and related to the specification

Name	Comments	Notification Mechanisms
TakeDown	All instances of this should be communicated to the Owner, and a record of them should be retained by the UGC site, for legal and audit reasons. The assertOwnership attribute represents a claim by the owner that they have rights over the content.	The EmailTakeDown field of the Owner element can be used for this. Some people think this needs to be a relatively immediate communication, so batching information up in log files is OK as long as those log files are communicated relatively frequently.
Quarantine	All instances of this should be communicated to the Owner, and a record of them should be retained by the UGC site, for legal and audit reasons. The assertOwnership attribute represents a claim by the owner that they have rights over the content.	The EmailQuarantine field of the Owner element can be used for this. Some people think this needs to be a relatively immediate communication, so batching information up in log files is OK as long as those log files are communicated relatively frequently.
LeaveUp	Records of LeaveUp actions need to be retained if the action has an expiry time (to allow for dealing with things before the expiration) or if the assertOwnership flag is true. The assertOwnership attribute represents a claim by the owner that they have rights over the content.	The EmailLeaveUp field of the Owner element can be used for this. Some instances of LeaveUp will need immediate attention, and some won't, so decisions about batching or not must be considered carefully.
ReportToOwner	This should be used for communicating things that are important, but don't have the explicit legal and contractual implications of TakeDown and Quarantine. The assertOwnership attribute represents a claim by the owner that they have rights over the content, even though they are choosing to do nothing at the moment.	Given that these are important, they could be sent to the Owner using EmailReportToOwner from the Owner element. Some UGC sites may choose to make timely deliveries of these batched up together, or even as specially flagged records in the general Log records
Log	This is used for things that are interesting, but that don't have great temporal value. For instance, if properly set up, this might provide	EmailLog from the Owner element should not be used for individual items, but only for batched bunches of entries.

	interesting data to mine for marketing or statistical studies. The assertOwnership attribute represents a claim by the owner that they have rights over the content, even though they are choosing to do nothing at the moment.	

When setting up the values for the ReportToOwner and Log elements, it is worthwhile to remember that they only need to contain information that will not be in the Notification element, since all information in the Notification element has to be delivered as part of these actions.

6.10 Multiple Recipients for Rules

The current specification does not support having different rules for different recipients (UGC sites, ISPs, etc) in one XML file, so it has to be handled as multiple files. There are some techniques that can be used to make this easier:

- Use a tool for generating the XML, rather than producing it by hand. MovieLabs will provide a tool that provides a simple version of this.
- If the rules have one recipient, use the SiteConcerned sub-elemnt.
- If the rules have more than one recipient, the Extra field in the Owner element can contain this information, which can then be extracted when the owner gets Logs, notifications, etc. The RuleName can be used similarly.

Please see the Future Features section for possible enhancements in this area.

7 XML SCHEMAS

There are 3 XSD files for the Content Recognition system:

- crGeneric.xsd -- schema for shared generic types
- rules.xsd -- schema for RuleList XML document (as defined in section 4.3)
- notification.xsd - schema for Notification XML document (as defined in section 4.4)

Current versions are available at http://www.movielabs.com/CRR

There are some implementation details to note:

- Special characters must be provided using the standard XML pre-defined entities
- Some examples of times can be found at:
  - duration conforms to http://www.w3.org/TR/xmlschema-2/#duration
  - dateTime conforms to http://www.w3.org/TR/xmlschema-2/#dateTime
- RuleList namespace-- http://www.movielabs.com/cr/rules
- Notification namespace-- http://www.movielabs.com/cr/notification
- ISAN IDs – If ISAN IDs are used
  - An additional namespace must be defined, e.g.:
    `xmlns:isan=http://www.isan.org/ISAN/isan`
  - An ISAN-based element must use the defned namespace, e.g. <isan:ISAN…/>
- Coral IDs – If Coral IDs are used
  - an additional namespace must be defined, e.g.:
    `xmlns:cor=http://www.coral-interop.org/arch/core/4-0`
  - A Coral-based element and all its inner elements must use the defined namespace, e.g.: <cor:resource.../>

8 REFERENCES

Technical, commercial, and social aspects of content protections

UGCPrinciples – Cross-industry statement on rights and responsibilities for content owners, ISPs, web sites, and consumers. http://www.ugcprinciples.com

EFF Fair Use Principles for User Generated Video Content – Another view of what policy should be. http://www.eff.org/issues/ip-and-free-speech/fair-use-principles-usergen

Digital Cinema Initiatives – Specifications for an open architecture to support digital cinema, including forensic marks. http://www.dcimovies.com

AACS – The Advanced Access Content System defines standards for various kinds of copy protection, including encryption and watermarks. http://www.aacsla.com

ACNS – The Automated Copyright Notice System is an XML format for automating responses to copyright infringement, found at http://mpto.unistudios.com/xml/

Application-level technical standards

ISAN – http://www.isan.org gives the background on ISAN, an international standard for assigning unique identifiers to an audiovisual work. ISO 15706-1 covers basic works and registration. ISO 15706-2 (also known as v-ISAN) deals with multiple versions of the same work. Both can be found at http://www.iso.org

GRid – Global Release Identifier, a unique identifier for music releases over electronic networks. http://www.ifpi.org/content/section_resources/grid.html

ISRC – International Standard Recording Code, international identification system for sound and music video recordings. http://www.ifpi.org/content/section_resources/isrc.html

Coral Consortium – A cross-industry specification for interoperable DRM systems, done with XML and providing good levels of abstraction for identifiers. http://www.coral-interop.org

Background Reading

Geofiltering – *Internet Geolocation and Media Services*, June 21, 2007. Report on the advantages, disadvantages, and realities of geofiltering systems. Prepared exclusively for MovieLabs members. Inquiries may be sent to info@movielabs.com.

Watermarking –*Digital Watermarking Survey*, TR-WM1, November 12, 2007. Report on the current state of the art in watermarking products. Prepared exclusively for MovieLabs members. Inquiries may be sent to info@movielabs.com.

Fingerprinting – MovieLabs has prepared several reports prepared exclusively for MovieLabs members. Inquiries may be sent to info@movielabs.com.

UUID – There is a good survey of techniques and sources for generating reliably UUIDs at http://en.wikipedia.org/wiki/UUID

Infrastructure Specifications

ISO Country Codes – ISO 3166-1 Alpha-2 code Elements - English - Country Names and Code Elements, found at http://www.iso.org/iso/english_country_names_and_code_elements

URI, URL – A top-level index page pointing to specifications is at http://www.w3.org/Addressing/

"URIs, Addressability, and the use of HTTP GET and POST" -- http://www.w3.org/2001/tag/doc/whenToUseGet.html

XML -- Extensible Markup Language is maintinaed by the W3C, and a good jumping-off page is http://www.w3.org/XML/ The W3C also maintains he specifications for the XML schema.

XML DATE/TIME -- The date and time representation specified in XML Schema Part 2: Datatypes http://www.w3.org/TR/xmlschema-2/#dateTime

XML DURATION -- The duration representation specified in XML Schema Part 2: Datatypes http://www.w3.org/TR/xmlschema-2/#duration

XML Character Escaping -- http://www.w3.org/TR/1998/REC-xml-19980210#dt-escape

9 APPENDIX I -- LICENSE

MOVIELABS CONTENT RECOGNITION RULES SPECIFICATION PATENT LICENSE

Purpose:
Motion Picture Laboratories, Inc. ("Movielabs") is the owner of certain patents and/or patent applications that may be infringed by products or services that are compliant with the Movielabs Content Recognition Rules Specification (the "Specification"). In order to facilitate the adoption and use of the Specification, Movielabs is willing to grant to any person or company that adopts or implements the Specification ("You") the following patent license subject to the terms and conditions below.

Acceptance by You:
No license will be granted to You unless You accept and agree to all of the terms and conditions of this Patent License. If You do not accept and agree to all of the terms and conditions, You shall not be entitled, expressly or impliedly, to any rights granted in this Patent License.

Licensed Patents:
As used here, "Licensed Patents" means all patents and patent applications owned by Movielabs that, in the absence of this Patent License, necessarily would be infringed by a compliant implementation of the Specification.

License:
In consideration of Your agreement to all of the terms and conditions of this Patent License, Movielabs grants You a personal, worldwide, nonexclusive, non-transferable, royalty-free license under the Licensed Patents to make, have made, use, import, offer for sale and sell products and/or services that are compliant with the Specification. This license excludes the right to grant sublicenses.

No license, either express or implied, or by operation of law, is granted by Movielabs with respect to any patent, patent application or other patent right except as specifically stated in this Patent License.

Limitations
Nothing in this Patent License shall be construed as:
(a) A warranty or representation by Movielabs as to the validity, scope or enforceability of any claim of any of the Licensed Patents; or
(b) A warranty or representation that any product or service made, used, sold or otherwise disposed of under this Patent License is or will be free from infringement of patents or any other intellectual property right of any third party; or
(c) A requirement that Movielabs will file any patent applications, secure any patent, or maintain any patent in force; or
(d) An obligation to bring or prosecute actions or suits against third parties for infringement of any of the Licensed Patents; or
(e) Conferring a right to use in advertising, publicity or otherwise any trademark or trade name of Movielabs, or any word or mark similar thereto.

Movielabs makes no representations, extends no warranties, either express or implied, and assumes no responsibility whatever with respect to the manufacture, sale, use or other disposition of any product.

THE LICENSED PATENTS ARE LICENSED AS IS AND MOVIELABS EXPRESSLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE AND SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, GENERAL, CONSEQUENTIAL OR OTHER DAMAGES (INCLUDING LOSS OF DATA OR PROFITS), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Term and Termination:
Unless terminated earlier by Movielabs as described below, this Patent License shall continue in force until the expiration of the last to expire of the Licensed Patents.

In the event that You or any of Your Affiliates (defined as having common majority ownership or control)

(i) owns or controls a patent, patent application or other patent right that necessarily would be infringed by a compliant implementation of the Specification, and

(ii) makes a claim, raises a counterclaim, or files a suit anywhere in the world, directly or indirectly, against Movielabs or any other party that alleges that making, selling, offering for sale, importing, or using a product or service that adopts or implements the Specification infringes such patent, patent application or other patent right,

then Movielabs may terminate this Patent License at any time upon written notice to You provided that such termination will be effective as of the earlier of the date of such claim or counterclaim or the filing of the applicable suit.

In the event that Your rights under this license or a similar license under the Licensed Patents have been terminated for cause by Movielabs, You shall not be entitled, expressly or impliedly, to any rights granted in this Patent License.

Assignment:
This Patent License may be assigned by Movielabs at any time. This Patent License may be assigned by You only with the prior written consent of Movielabs. Any purported assignment or transfer of this Patent License or any rights hereunder by You without the consent of Movielabs shall be void (without affecting any other licenses or rights hereunder).

Governing Law:
This Patent License shall be interpreted and construed in accordance with the laws of the State of California in the United States without regard to any conflicts of laws rules or principles.

10 APPENDIX II – FUTURE FEATURES

There are several obvious areas for extension of this specification. There are, loosely speaking, 5 or so categories – miscellaneous new features, improvement of the specification document itself, content recognition systems, viewing-time actions, and the relationship between a rights holder and a UGC site.

- Miscellaneous features
  - Graduated response is working as a prevention/deterrence mechanism. An action for something like 'increment graduated response counter for originator' would be useful; it would bring in another unspecified external system, though.
  - Hierarchical groups
    - Currently an episode can be a member of a season group and of a series group. It might be more natural to think of an episode as a member of a season group, which itself is a member of a series group.
    - URI-based groups could be extended to manage hierarchies
  - Other grouping mechanisms, including groups based on coral resource-chain elements
  - Some have expressed a need for there to be way of excluding certain segments as part of the underlying Asset as well as part of a Rule
  - ISAN versions might conceivably be helpful for certain kinds of content replacement operations.
  - SectionMatched is specified in seconds, since that's the limit for most matching systems right now. Eventually, it might want to be specified using a TimecodeRange
  - There is some interest in extending the rules to cover different classes of devices.
  - In some cases, there will be fingerprints not only of an original asset, but of uploaded copies of it as well produced, for example, by the distribution of the copy over multiple generations to multiple sites. It would be useful if these could all be tied back to the 'master' original asset somehow.
  - Trailers can be covered (see the use cases), but the mechanism has some failings. There have been requests for explicit management of trailers, some of which involve adding operators beyond the AND implicit in the detection criteria and the ordered list or OR clauses represented by the individual rules, and some of which require feeding the results of one rule as parameters into another.
  - Whitelist management could be expanded on and formalized.
  - Although the notion of a rating (MPAA, BBFC, etc) is usually found as part of descriptive metadata, it may have some use in these rules, and could be added.

- Allowing an AssetList to refer to multiple templates; this requires careful definition of how conflicts in the templates get resolved.
- Allowing template ID types other than UUID

- The specification document itself
  - Use cases have been requested for MatchThreshold, SectionMatched, DCI watermark detection, and use in a watermark-based identification system.
  - In order to ensure formal extensibility, xsd:any could be added in the schemas for Actions and DetectionCriteria.
  - Alternate expressions of the rules – RDF and OWL have both been suggested as good ways of expressing the rules. We chose XML for the first version because it is simpler and has wider acceptance, and currently more of our target audience of decision makers and implementers are literate in XML than in RDF or OWL.
  - The applicability of a similar specification to peer to peer networks should be investigated
  - Some parties have expressed interest in the availability of formal test suites and a simple but complete UI-based tool for generating rules
  - More formalized error handling would be very helpful.
  - Some of the items in the definitions section need improving.
- Content recognition systems
  - The current specification is deliberately loose about requirements on recognition systems. This is appropriate for the current state of the market and technology, but ought to be tightened up eventually, with particular urgency for the concept of matchThreshold.
  - Length and position of match within a site asset to be returned in the Notification element – It is useful when dealing with false positives and more complex business rules to know each time at which a particular original asset occurs in a site asset, and for how long. This requires either dictating or allowing the specification of continuity criteria (e.g. if 10 4.5 sec segments match, and then one doesn't, and then 10 more do, is that two 45 second matches, or one 90 second match?) SectionMatched starts down his path.
  - Original assets that contain multiple fingerprinted works – if an original asset is a composite of other original assets, the composite original and the individual originals will all generate fingerprint matches. This will generate a chatty set of results currently, but will still trigger the appropriate actions. There is currently no way to determine whether or not a site asset matches an entire composite original.
  - Original assets that are separate but contain identical (from the point of view of the recognition system) material. This might happen with a movie and its trailers, for instance. Current recognition systems will match both, and it may be the case that the desired behavior cannot be managed with two separate sets of rules.

- Some parties have expressed interest in detection rules for unacceptable advertising; this has the same technical knots as vendor-specific watermarks, since there is no real standard (or even description) for such a system.
- Some assets in the future might have components that can be matched that are neither audio nor video. It is obvious where to put this, but not currently obvious what it means.
- Watermarks
  - Vendor-specific watermarks – version 1.0 matches against some standard and well-defined watermarks. There are many vendors providing B2B and B2C forensic marks. General rules to detect specific vendors' marks require having a watermark/domain pair, which is a similar problem to having multiple types of IDs (addressed in this spec and by the Coral system-specifier element), which would be a better solution than a partial enumeration of marks and syntax from some number of currently active vendors. There are also levels of detection available – e.g. detect existence vs detect varying levels of detail.
  - The detected watermarks criterion supports DCI forensic watermark, AACS consumer mark, and AACS theatrical – no home use. Are there any other standard watermarks worth detecting, e.g. some standard copyright flag, or an overriding mark allowing unrestricted use?
- The MatchedComponents attribute of a rule may not be implementable by all vendors. In a world with multiple audio tracks in uploaded content, it will probably have to be replaced with an element – more complex, and not worth doing for this version.

- Viewing-time actions
  - More elaborate management of ads around infringing content is an interesting area.
  - Some reviewers have suggested having the provision of a snippet of required HTML as one of the actions
  - Some specs for content distribution (e.g. CableLabs VOD and some recognition vendors) explicitly talk about revenue share percentages. Currently, this spec is not an appropriate place for that, but maybe in the future it could be.
  - There is a set of actions around what can happen to content once it is available on a UGC site. Currently, advertising is the only one mentioned; some others are allowing user comments and allowing external linking.
  - AlternateContent should be expanded or tightened up. It might allow, for example, inclusion of XML snippets, HTML that contain images and links, or a standard RSS feed. The desire of content owners to present their content the way they want to, the desire of UGC sites to retain control of the layout and appearance of their sites, and the desire of consumers for their legitimate uploaded items to be seen all have to be carefully balanced.

- An action of 'No Ads' has been proposed.

- Rights holders and UGC sites
  - Formalization of reports to the content owner. The current version only specifies a request for the information, with no details on contents or formats. This would be useful for information about both uploads (successful and unsuccessful) and access to allowed content (including the rules applied at access time.)
  - Currently, if a rights holder wants to have different rules for different UGC sites, each UGC site must get its own separate XML file. Some rights holders might want a single source for all rules for a piece of content, independent of the recipients of those rules. This could be done as a query to a (possibly centralized) repository, or by including site-specific directives or grouping in the XML.
  - It would be helpful to have multiple recipient entities in the XML, for distribution and tracking purposes. Currently a single one can be kept in the SiteConcerned field of the RuleList. Multiple ones can be kept in the Extra field of the owner, since it's just informational, or managed as part of the RuleList name. The type of recipient (UGC site, ISP, etc) would also be useful.
  - The notion of 'ownership' is quite simple, and does not cover cases where one entity has rights to a soundtrack and another has rights to the video, for example. For the current specification, that level of detail needs to be resolved as a business negotiation before the rules are crafted – it is easier, for example, for a site to send a revenue share to one recipient who then divides it appropriately rather than to multiple recipients who may have changing relationships with each other. Future versions of the specification may consider addressing this more formally.
  - Also for the notion of ownership, some parties have expressed interest in having a hierarchy of owners specified, as a potential way of automating more conflict resolution.
  - The Geography element on the Owner element is redundant with the geography elements in the Action elements, and is really now just a help for dealing with certain kinds of rule conflicts. Is there a way to make it more useful?
  - The owner element could have time windows associated with it.