Taxonomy of Control Mechanisms

Florian Schuetz
Advised by Manuel Hilty, Alexander Pretschner

Information Security, ETH Zurich, Switzerland
fschuetz@ethz.ch

August 3, 2006

Abstract

Modern protection requirements extend beyond classical access control models. A wide variety of control mechanism has been developed to meet those requirements. This report defines classification criteria and provides a survey about modern control mechanisms.

1 Introduction

How digital data may be used and distributed is a concern in many areas of information security. Data protection is about controlling the use of sensitive personal data to protect people's privacy. In digital rights management (DRM), usage of data is controlled in order to protect intellectual property rights. The use of confidential business or military data often needs to be controlled as well. Embracing all these areas, we define usage control (UC) as an extension of access control (AC) in that control extends not only to who may access which data, but also to how the data may or may not be used or distributed afterward.

In addition to access control requirements, usage control introduces requirements about the further usage of data once it has been given to somebody else, so-called obligations. Examples include "data must not be further distributed" and "data must be deleted within 30 days". As introduced in [44], we classify obligations according to their controllability and observability. Dedicated mechanisms may be used for controlling and observing the fulfillment of obligations, and thus help with the enforcement of usage control requirements. Such mechanisms can for example be found in the area of digital rights management (DRM) [26].

Control mechanisms exist in a wide variety. Most control mechanisms are tailored toward special tasks such as copy protection or secure distribution. It is unclear whether those mechanisms are applicable in other contexts, for example to solve privacy issues. This classification is constructed such, that the applicability of a mechanism is easily decidable.

Definitions of control mechanisms are rare. If definitions are made, they are application specific, which only allows discussion of mechanisms with similar properties. Those definitions do not provide enough flexibility to discuss arbitrary mechanisms, be it a simple copy protection scheme or a sophisticated client-server system. The definition provided in this report is applicable to

arbitrary protection schemes. Only such a definition allows comparison and classification of diverse systems.

To judge the capabilities of different control mechanisms, suitable criteria for classification must be decided. The number of criteria must be as small as possible while still giving a broad and deep insight into the capabilities and limitations of the control mechanism. Also non-functional properties must be considered, since they are key criteria for using a mechanism in everyday business.

This survey is aimed at providing a comprehensive, quick overview about properties and applicability of control mechanisms. It provides an insight into the large field of control mechanisms. Preselection of potentially interesting mechanisms can easily be done based on usage requirements. Further this survey reveals the capabilities of today's mechanisms. It can therefore be used to decide which fields need some more efforts and research.

The remainder of this report is organized as follows. Section 2 analyzes important aspects of usage and how it can be controlled. Based on this, section 3 works out a definition of "control mechanism". Section 4 discusses the criteria used to classify control mechanisms. The researched mechanisms are presented in section 5.

2 About Usage Control

Some of the criteria that can be used in a taxonomy about control mechanisms are known from classical access control. However, some issues peculiar to usage control arise. It is therefore necessary to understand the subtleties of usage and how it can be controlled.

Usage requirements are the core concept behind usage control since they formulate how data may be used. They can be formalized as formulae interpreted on execution traces. Consider a system as a state machine. The possible transitions are defined by what forms of usage can occur, the first aspect of usage requirements. Section 2.1 discusses the characteristics of different forms of usage. Conditions, the second aspect of usage requirements, impose when a transition is valid. They are discussed in section 2.2. A control mechanism hinders that a system makes any invalid transition. In other words, it enforces usage requirements.

Finally, section 2.3 discusses how usage requirements can be enforced. While access control limits enforcement to whether access is granted or not, usage control allows more subtle considerations.

2.1 Usage

Different types of usage exist. Application of a filter to improve an image, displaying an image on the screen, copying a file to a memory stick or sending it by email, gathering metrics from source code and execution of a program are all different types of usage. It is important to understand what the differences are and what types of usage exist. Only this allows a comparison of what can be controlled by different mechanisms. The categories should be as comprehensive as possible to allow efficient comparison.

Consider the examples above. Two types of usage can be distinguished: One that interprets data and one that does not. Data is interpret, whenever the result of usage is influenced by the meaning of the data. For example, when copying a document to a memory stick, the data is replicated at on the memory stick, regardless of how the bit pattern looks like. The data is not interpret. On the other hand, when playing a movie, the color of the pixels on the screen is dependent on the bit pattern of the data, thus data is interpret. We define whitebox usage as usage that interprets data and blackbox usage as usage that does not interpret it. Copying a file on a memory stick

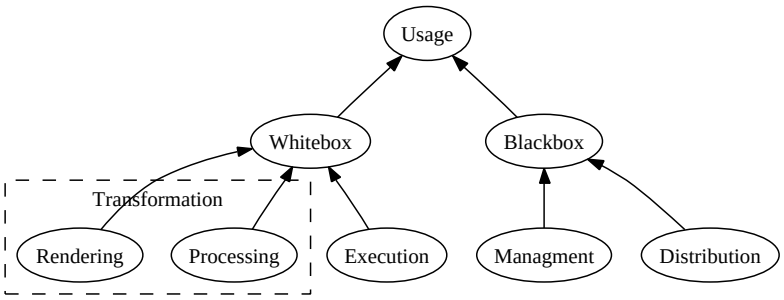

Figure 1: Usage Classes

or emailing it is black box usage. Filtering an image, displaying it on the screen and executing a program are whitebox usage.

The distinction of blackbox and whitebox usage is too coarse for differentiated considerations. Blackbox usage of data can be divided into two further categories: Distribution and Management. Management deals with copying, storage and deletion of data. Copying a file on a memory stick is management. Distribution deals with sending data. Sending data means that the data is transfered from one data consumer to another. Sending a file to a friend by email is distribution. It would not be distribution however, if the file was sent to oneself.

Three distinct forms of whitebox usage can be differentiated: Execution, processing and rendering. Running a program is execution. Execution manipulates the system state. A further aspect of whitebox usage is about transforming objects. The calculation of metrics for example transforms the code object into the metrics object. Processing does alter the data, while not changing its physical representation. Rendering does change the physical representation of data. The application of a filter to an image is processing. Displaying that image on the screen or printing it is rendering.

Procedures in a system seldom relate to one of those categories alone. If Alice stores the final version of a business report in a company database on a server, then this is distribution and management, in this order. Another, more complex example is editing a file with a word processor. When editing, keystrokes are directly displayed on the screen and the data in memory are altered. This can be modeled by sequencing processing and rendering activities. This shows that the processes occurring in a real system can lead to complex combinations of these usage categories, but all of them can be modeled.

For the rest of this document we will refer to the usage categories discussed above as *Usage Classes*. The class concept will be used throughout this report. As in programming, a class intensionally defines a set of instances. How an instance looks like depends on the discussed topic. Considering the classes above, printing a document is an instance of rendering.

The use of the class concept allows a flexible discussion about usage. Going up in the class hierarchy results in generalization and allows discussing about common properties in an easy and understandable way. Going down in the class diagram allows specialization. The diagram displayed in Figure 1 is by no means complete. It can be extended at the bottom. For example a “print” class which inherits from rendering could be added. This is important, since control mechanisms might not support all operations in a usage class. In that case subclasses can be introduced to reason about the capabilities.

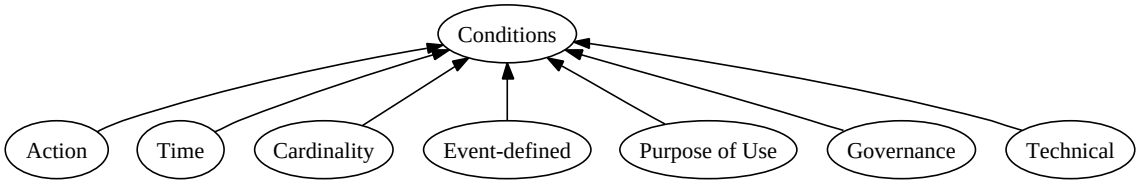

Figure 2: Usage Conditions

2.2 Conditions

Conditions narrow down usage. They affect how data may be used. A control mechanism evaluates if all conditions hold and only then allows performing a certain action. If one of the conditions does not hold, it will counteract as discussed in section 2.3.

Action conditions describe which actions must be performed to enable usage. Examples are that a fee must be paid before listening to a piece of music. Further conditions are time, cardinality, event-defined, purpose of use, governance and technical conditions. These are always bound to an action condition. Conditions can be combined to form complex statements in a policy. For example, rental can be expressed by combining action conditions, time conditions and purpose of use: "Video file must be *deleted* after *7 days* and may *not be shown in public*."

Time conditions limit actions to certain times or dates. There are two distinct forms of time conditions: Period of time and point in time. Period of time for example states "document must be deleted within 7 days". An example for point in time is "start backup at 23:00". Time conditions may be combined with any other usage requirement.

Cardinality conditions allow to limit the number of usages or demand a certain amount of actions to be executed. Cardinality conditions can be combined with time, event-defined and technical conditions as well as purpose of use. Examples for cardinality constraints are: "Movie may only be played once", "watch preview at most twice" or "do not show in public more than 3 times".

Events are always raised by an action. Event-defined conditions are of the form "if *action* ..." and can be combined with any other usage requirement. An example for an event-defined condition that requires an action is: "If the author releases a new version then update immediately." Also changes of licenses such as "content may not be quoted until the author explicitly states otherwise" fit into this category.

The usage of an object can be dependent on the purpose of use. Objects that are labeled "For personal use only" may not be used in a business context.

Governance conditions may demand compliance to a certain standard. A license could state "Only use when organization complies to the Sarbanes-Oxley Act (SOX)".

Technical conditions demand the technical environment to have certain properties. A license could state "Only use on devices compliant to the Common Criteria".

2.3 Enforcement

If a user tries to perform an action that violates any condition imposed on data, then the control mechanism must counteract. It can do so in different ways. This is what we call enforcement.

The easiest way of enforcing conditions is inhibiting operations. Consider a user playing a video

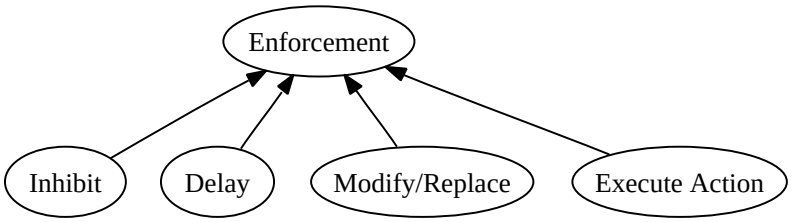

Figure 3: Enforcement

file. Inhibiting usage means that every attempt to play the file fails completely. For example the play button might be deactivated.

Another possibility is to delay actions until every condition is fulfilled. For example, as soon as the user pays a fee, the file is played. Note that delaying actions is only valid for a finite time period.

A content distributor also might choose to provide the user with a low quality preview if not all conditions are fulfilled. This can be done by modifying the effect of usage or by replacing the intended action. The video might for example be run through a filter that inserts noise or there might be a second low quality video encapsulated in the data that is played instead of the full version.

The last possibility is that the violation of a condition leads to an action. For example the data provider might be informed if a user tries to play the video but does not satisfy all conditions.

Action execution needs to be handled with care, since it exists a very similar use case that is not controllable. Consider that a data provider named Bob sends a music file to Alice. Bob demands, that Alice does not make any recordings of it. There is no way in controlling whether Alice places a tape recorder in front of her speakers and violates the policy. This can be dealt with observation mechanisms as introduced in [44]

The considerations above lead to the following enforcement classes: *Inhibit*, *Delay*, *Modify/Replace* and *Execute Action*.

3 Definition of Control Mechanism

To compare control mechanisms, the exact meaning of the term "Control Mechanism" needs to be fixed. An expedient definition must be applicable to arbitrary protection systems. Further it does not suffice to state what a control mechanism does. The definition must express what a control mechanism is. To come up with such a definition, the environment control mechanisms work in must be carefully examined.

Figure 4 depicts an abstract usage control environment. A data provider prepares the data it wants to distribute. Devices are used to make the data controllable. Then the data is distributed by media, for example on a CD or over a network.

The data consumer that received the data can do usage on his devices according to the conditions set by the data provider. Some invariable conditions that apply to any protected data are directly built in the control mechanism. Since this does not allow giving different rights on different data, often the rules will be provided with licenses.

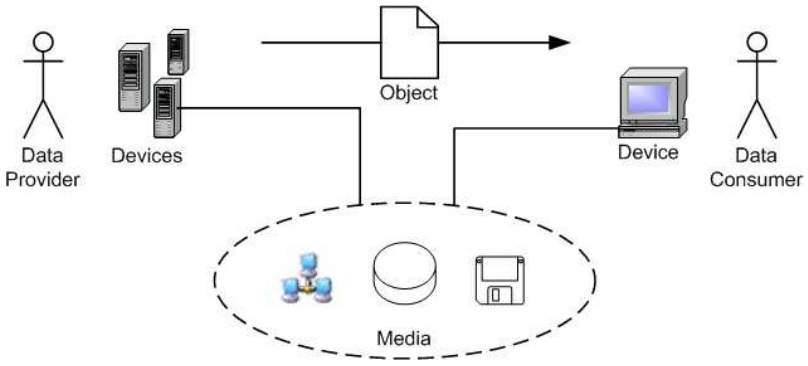

Figure 4: Usage Control Environment

A license is an agreement between a data consumer and the data provider which contains usage requirements that the data consumer committed on. It binds the data provider, the data consumer and policies to an object. To be able to use that object, the data consumer must fully agree on the content of the license. A license can be obtained through negotiation, since a data provider may have some leeway.

The data that need protection are called objects. We use the concept of classes as introduced in section 2.1 to discuss groups of objects with common properties. An object therefore is an instance of a *data class*. Many different data classes exist. Examples are "audiovisual content" or "Microsoft Office Documents".

Please notice, that objects are independent of their representation. The analog signal that is transmitted to speakers when playing music is just another representation of the same object stored on the player's disk.

Objects reside on media. A medium is anything that carries data but cannot process it. It is important to discuss media, since a control mechanism may apply control to media rather than to objects themselves. Consider for example a copy-protected CD that carries a video game. It is unlikely that someone tries to bring the game in every possible game state while recording the instructions sent to the CPU. Thus even if the game data itself is not protected at all it can not leave the CD.

Usage of objects is done by devices. Devices are implementations of algorithms with execution, data processing, rendering, distribution and/or management capabilities. Usage is done by applying capabilities on objects. A device can be a hardware mp3 player, a computer system, a virtual machine or even just a specialized software player. A DVD player is a device that has capabilities to play movies, select titles and many more.

Taking into consideration the topics discussed in chapter 2 this leads to the following definition:

Definition 1 *A* control mechanism *is an implementation of algorithms, a dissemination format / medium or a combination of both that restricts the usage of objects and/or enforces certain actions to be executed at a certain point in time.*

An example for a control mechanism that is a dissemination format is encryption. Assuming cryptography to be perfect, the content is neither revealed, nor can it be modified. The classic example for a control mechanism that is a dissemination medium are early CD copy protection

schemes. These schemes modified sectors on the CD. When playing the CD, personal computers skip the bad sectors. But when one tried to copy the CD, the software detected the errors and aborted. The "Extended Copy Protection (XCP)" used by Sony on some audio CDs is an example of a control mechanism that is a combination of a dissemination format and the implementation of an algorithm. The CD contains both, the audio tracks and a data track. The data track is ignored by music CD players. However, a PC that uses the autostart feature provided in the Microsoft Windows operating system will silently launch an installer that installs a control mechanism on the system.

Algorithms may be implemented in hardware, software or as a combination of both. Consider the Content Scrambling System (CSS) which protects DVDs. The control mechanism that does the validation and decoding is implemented in hardware in DVD players as well as in software in players for the personal computer.

4 Criteria for Classification of Control Mechanisms

The criteria for classification can be divided in three parts: Applicability, implementation and non-functional properties. Applicability deals with whether a mechanism can be applied to a certain problem. It covers what objects can be protected, how they can be protected and how protection is enforced. Implementation deals with how different aspects of the mechanism are designed and embedded into systems. Non-functional properties provide an overview about the performance, security and the costs of the mechanism.

4.1 Applicability

Usage Requirements

Usage

Covers which of the four usage classes *management*, *processing*, *distribution* and *rendering* the mechanism can control. The limitations of control are not considered here. For example, a mechanism that inhibits sending objects by email would get the attribute "distribution", even if it is not able to control transfers over ftp.

This is because every form of control is limited as long as something can be done with the object. A movie can always be captured with a camcorder from the screen.

Conditions

Covers which conditions are enforceable. Conditions subdue usage as introduced in 2.2. Any of the conditions can either be supported fully, partially or not at all.

Enforcement

Covers which of the usage enforcement classes defined in section 2.3 the mechanism supports.

Limitations of Enforcement

Covers under which conditions the mechanism releases objects from its control. A short text will explain when content is released and if any measures are taken to allow observing it.

Data

Covers which kind of data are controllable by the mechanism. This can be specific file types or data classes like "audiovisual content".

Environment

A control mechanism may be limited to certain environments. Organizational and technical environments can be differentiated.

Technical

Covers which class of of media and/or devices are required for the mechanism to work. Further it covers which are prohibited and would cause the mechanism to fail. A control mechanism might for example need an Active Directory to validate subjects.

Organizational

Covers which organizational measures need to be complied with for the mechanism to work as expected. If for example it is mandatory that certification is conducted according to a standard, the mechanism may not work properly if certificates can be created by the employee himself.

4.2 Implementation

Under implementation every aspect that directly influences the power and the behavior of the control mechanism is covered.

Layer of Embedding

A control mechanism can be implemented at different layers. The highest layer is the *Application* layer. Control mechanisms at the application layer run as userspace programs and do not need any modifications of the operating system. Mechanisms at the *Operating System* layer are either directly included into the kernel or run as privileged modules. They do have direct access to resources that are controlled by the operating system. The lowest layer of embedding is the *Hardware* layer.

The functionality of control mechanism may be distributed over several layers. Any combination is possible.

Protection Layer

Covers on which layer the mechanism protects objects. The protection may be done on the *data layer*, the *media layer*, the *device layer* or any combination thereof.

Data Layer protection directly protects the content by choosing an appropriate dissemination format such as encryption or watermarking. The security of these techniques is discussed in [26].

Protection on the media layer is achieved by choosing an appropriate dissemination medium. Early CD copy protection for example is a mechanism that operates on the media layer. The data itself is not touched at all.

Device layer protection is similar to protection at the media layer. The device that stores the content is designed such that it mediates all accesses to data. The data cannot be extracted without proper rights. An example is the Apple iPod when used with iTunes. Once protected songs are copied to the iPod they can not be reclaimed.

Model

The model of a control mechanism can be either *Local* or *Client-Server*. A local control mechanism works without any online connection. A client-server mechanism needs at least once an online connection. This implies that a mechanism is client-server even if it just fetches the license at the first time the protected object is accessed and then works without connection.

Different connection models are possible for client-server mechanisms. Possibilities are *Online* or *Partially Online*. Online means that the connection needs to be available permanently, otherwise usage is not possible. A control mechanism that supports pay-per-time and does the billing on a remote server, needs to be online.

Partially online means that the mechanism allows usage when offline, but needs an online connection from time to time. An example would be a mechanism protecting documents. This mechanism could check every three weeks on a remote server, if the license is still valid or has been revoked.

Feedback Support

A control mechanism may support feedback to inform the data provider about certain events. It is obvious, that this requires an online conection. Partially online mechanisms must use logging facilities. They will provide feedback as soon as an online connection becomes available. However, when discussing online mechanisms, logging capabilities and feedback support are not coherent.

Logging Capabilities

A control mechanism may provide logging facilities. These may serve the purpose of control or observability. A control mechanism may log the number of usages, to be able to control cardinality constraints. On the other hand, logging may allow a control mechanism to record license violation. Either those information can be provided to feedback mechanisms for reporting or they can be stored for future evaluation. In this case, logging is used for observability. Please note, that logging is independant of whether a mechanism supports online connections or not.

License Management

Which usage requirements can be enforced by a mechanism is – among other factors – dependent on the license and the supported rights expression language. The supported rights expression language defines which usage requirements are enforceable. Where the license is stored has implications on the functionality and reliability of the control mechanism.

Storage Model

The license which describes usage restrictions on objects can be *stored with content*, in a *local repository* or in a *remote repository*.

Negotiation

The exact content of the license can either be given directionally by the *Data Provider* or negotiated in *Bidirectional* communication.

Each form of negotiation can either be *Automated* or *Manual*. A negotiation is automated if the subject acquiring the license has no direct mean to influence the process. Manual

on the other hand means that a subject actively has to agree to a license, for example by clicking an "Ok"-Button.

Target of License

A license may give rights to different targets. Possible targets are *Subjects* or *Devices*. For example a license may state that subjects are not allowed to copy objects or it may state that devices are not allowed to perform the action copy. While the effect – assuming all entities to play fair – is the same, the initial position is not.

Group Support

Cover whether the license is applicable to groups or not. If a license does not support grouping, there needs to be one license for each target. Grouping allows more flexibility and the reuse of licenses. It is much more convenient to give the modification right to every subject in the group "Managers" and drastically reduces the administration effort.

Default Behavior

A License is using either the *Whitelist*, the *Blacklist* or a *Fixed* approach. Using the whitelist approach means that every usage not covered by the content of the license is denied. Blacklist is the opposite, every usage that is not inhibited by the license content is allowed. A license is said to have fixed default behavior, if for every right other defaults exist. For example, the play right could be granted by default, while the record right is denied by default.

Rights Expression Language

This attribute mentions which rights expression languages are supported by the license. The license may provide *full* or *partial* support for any of the supported languages.

4.3 Non-functional Properties

None of the systems considered in this report were available for testing. The data that caused the classification are collected from documentation and reviews available on the Internet. Therefore there is no finer partition than into low, medium or high.

Performance

Covers how fast the control mechanism performs. A mechanism that controls multimedia content needs to perform fast. But also for non-entertainment oriented mechanisms performance is an important factor. Nobody likes waiting 5 minutes until a document opens, just because checking the license takes that long. Possible values are *Low*, *Medium* or *High*.

Reliability

Reliability is concerned with how stable a system runs. Does it crash often? Is there a high probability that checking a valid license fails? Possible values are *Low*, *Medium* or *High*.

Usability

Usability of a control mechanism will decide whether it is accepted and adopted or not. We distinguish between the consumer and the distributor. Possible values are *Intuitive*, *Learnable* or *Complicated*.

Costs

Direct Costs

Direct costs are assigned the values *Low*, *Medium* or *High*. There is a distinction between data consumer and data provider. For a data provider a mechanism that costs $5000 is cheap, while for a consumer this is very expensive.

Implementation Expenditure

Implementation expenditure covers how many systems need to be changed or newly obtained. It also covers if business processes need to be adapted to the new control mechanism. Possible values are *Low*, *Medium* or *High*.

Security

Security can be *Low*, *Medium* or *High*. Security is measured by how easy it is to break a mechanism. A mechanism that operates at the application level will usually be easier to break than a mechanism operating at the hardware layer.

Controllability

Controllability sums up how much power a mechanism has. The values are gained by combining the results from the applicability and implementation criteria. For example, a mechanism at the hardware layer which only allows rendering of data on digital outputs is stronger than a mechanism that can not distinguish between analog and digital outputs. This is, because digital outputs allow encrypting a signal until released as a sound wave, for example. An analog output can not be encrypted and an attacker might grab a high quality signal.

Possible values are *Low*, *Medium* or *High*.

5 Classification of Control Mechanisms

In this report ten control mechanisms are classified according to the criteria defined in section 4. Two systems – AACS and RMS – are discussed in detail. These two control mechanisms differ in many aspects, which allows for a good insight on how classification has to be done.

The remaining mechanisms are briefly introduced and particular characteristics are highlighted. References to further information are provided. The interested reader can use them to validate the classification results and gain deeper insights into the mechanism.

A tabular overview about all the control mechanisms examined in this report can be found in Appendix A. The symbols in use are explained in Table 1.

5.1 Advanced Access Content System (AACS)

The "Advanced Access Content System (AACS)" is a control mechanism to protect next generation optical discs. It has been adopted for Blu-ray discs and HD DVD. The specification [21] is organized in several books that are concerned with different uses of AACS. It is published by the "Advanced Access Content System License Authority LLC (AACS LA)". Currently only interim licenses are available. AACS LA claims that final version will provide additional features. Announced features are considered in this classification.

AACS is intended to replace today's entertainment systems allowing distributors to define fine grained usage requirements. It aims at eliminating illegal distribution of content. Unlike many

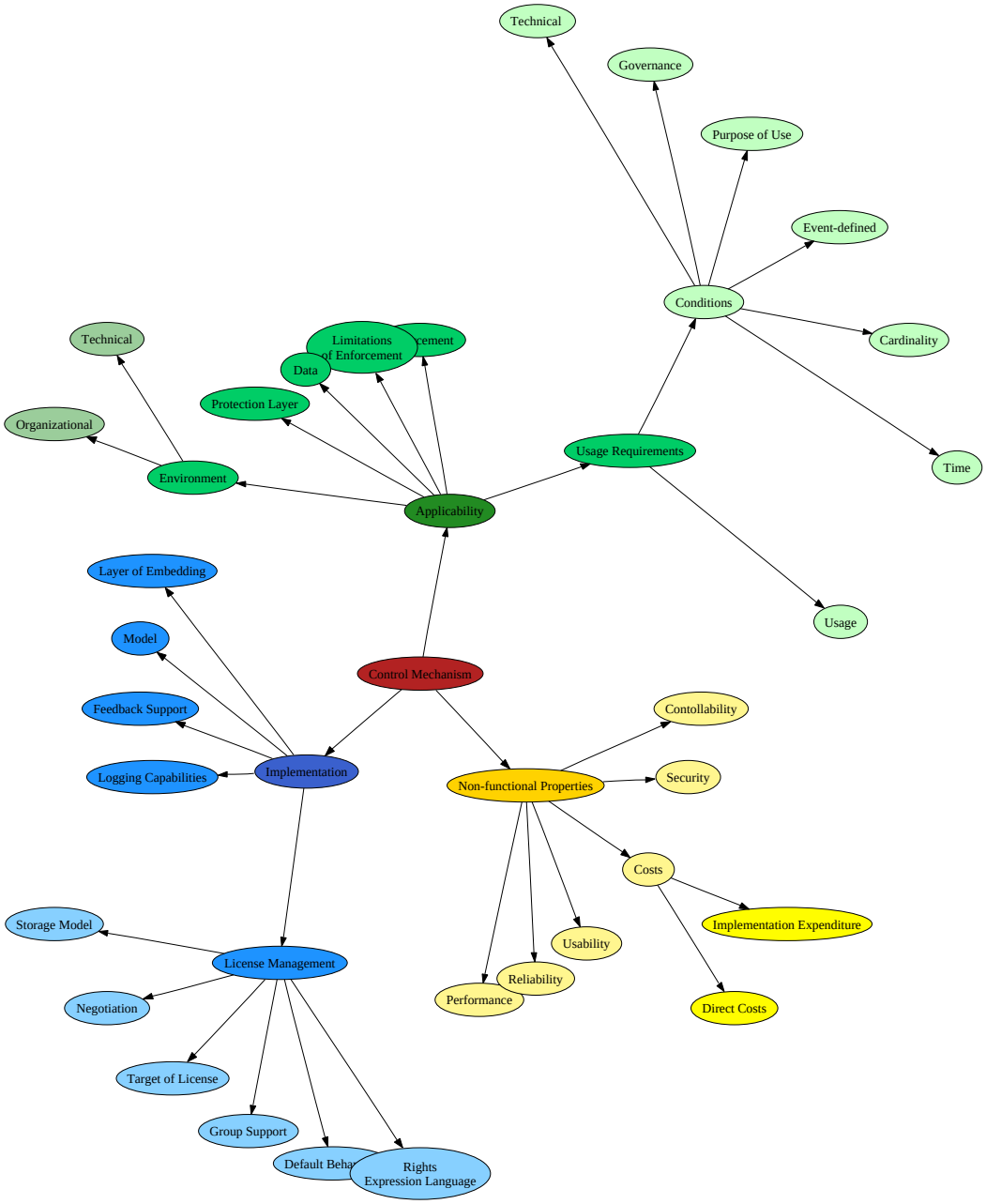

Figure 5: Classification Criteria

other DRM mechanisms, AACS supports enhanced usage scenarios, for example storage of content on a media server.

Every AACS drive has its own keyset. Keysets are distributed by the AACS LA. Further, there exists a "Media Key Block" on every medium. The device keys and the Media Key Block are needed to calculate the decryption key, which is used to decrypt the data. The Media Key Block is generated by the licensing entity and can be designed such that not every set of device keys can calculate the decryption key.

Further information can be found in [34, 33, 12, 23, 24].

5.1.1 Applicability

Usage Requirements

Usage

AACS can control processing, rendering, management and distribution.

Processing and rendering are controlled by the usage rules provided with the license. Management is supported in such that content can not be extracted from the medium. The device blocks any invalid access. If the device disposes of a robust online connection, enhanced usage scenarios like managed copy become available. This allows for example to store a copy of a movie on a media server. Distribution of content is only possible if the distribution right is given. Any AACS device will refuse sending objects if the right is missing.

Conditions

AACS supports action, time, cardinality and event-defined conditions. If an online connection is present, the need to contact the content provider before getting the right to make a managed copy can be seen as an action condition. Cardinality conditions are only supported for recording and copying functionality. Time conditions can express periods of time. There is no support for the expression of points in time. AACS supports device and content revocation. These are event-defined conditions. Other event-defined conditions are not supported.

Enforcement

AACS supports the enforcement classes inhibit and modify/replace. Unauthorized actions are inhibited. Rendering on analog outputs can be inhibited or the quality of the playback can be reduced. Further, the "Image Constraint Token" allows a content provider to render low quality or customized content on analog and digital outputs, if certain conditions are not satisfied.

Limitations of Enforcement

Control is naturally limited by the fact, that one might build a device, that directly accesses the media on the physical layer. With such a device content may be copied and distributed. The copy can be played just like the original.

Once rendered, the interim license does not specify mechanisms that allow tracking of content. According to [12] the final version of the license includes audio watermarking which could enable a certain degree of observability.

Making a non-AACS copy – which is supported by the managed copy function – releases the object from the control of the mechanism.

Data

AACS can protect audiovisual content.

Environment

Technical

To consume AACS protected content the consumer needs an AACS compliant device. If a user wants to record content, only AACS compliant media can be used. Storage on other media is prevented by the control mechanism.

Organizational

The content provider must cooperate with a licensing entity and a trusted replicator.

5.1.2 Implementation

Layer of Embedding

AACS is designed for implementation in hardware as well as software.

Protection Layer

AACS encrypts objects stored on the media. The cryptographic methods used are explained in [22]. Extensions for prerecorded and recordable content are explained in [23] and [24].

In addition, AACS compliant devices block any illegitimate access to the data.

Model

AACS works locally, without the need for an online connection. Online connection nevertheless are supported and allow enhanced usage scenarios.

Feedback Support

In [23] it is said that "a home video server might connect with a service provider to obtain authorization to make a protected local copy of a given pre-recorded title for "jukebox purposes". This allows the content provider to gain evidence about how content is used and may therefore be seen as feedback.

Logging Capabilities

Logging is not supported.

License Management

Storage Model

The usage rules are provided by the content provider and are packaged with the content by the license replicator.

Negotiation

The content of a license is given by the data provider. There is no support for negotiation.

Target of License

AACS licenses specify what a device can do with the content. The target of the license are therefore the devices.

Group Support

Grouping of devices can be done by equipping them with the same keyset.

Default Behavior

AACS follows the whitelist approach with the exception of the "play" right. The play right is granted by default and must be restricted explicitly if desired.

Rights Expression Language

The specification does not mention which rights expression language is used by AACS.

5.1.3 Non-functional Properties

Performance

Performance is high, since the control mechanism allows fluent playback of high quality video and audio content.

Reliability

The system needs to function very reliable, otherwise customers would not buy it. One can assume that this goal is achieved by AACS devices.

Usability

The system is intuitive for the content provider and the consumer. The consumer just needs to insert the disk and press play. The content provider only needs to provide the content and the usage rules, the rest of the distribution process is handled by specialized third parties.

Costs

Direct Costs

Nothing is said about direct costs by the AACS LA. However, one can assume the costs to be high for the content provider. The costs for the consumer will also be high, since he needs to buy new hi-tech devices.

Implementation Expenditure

The implementation expenditure is medium for the provider, which needs to adjust its production and distribution strategies. For the consumer it is high, since he needs to replace all the older devices by AACS compliant devices to profit from high quality content.

Security

AACS provides first class security features such as strong cryptography, drive authentication and digital signatures. Decryption keys are directly stored in the hardware. Assuming the critical parts to be tamper resistant, this provides a high degree of security.

Controllability

Controllability is high. According to AACS LA, from the year 2010 only digital outputs are supported. Especially after this "Analog Sunset" objects are controllable along any path through the system until rendered.

5.2 RMS – Rights Management Services

The "Rights Management Services" is an operating system module by Microsoft. It is integrated in Windows Vista. Microsoft also provides the module as a download for Windows Server 2003. Microsoft provides a developer kit which allows including RMS functionality in own programs. Currently the only RMS enabled application on the market is Microsoft Office 2003. Further Microsoft provides a Service Pack to enable RMS for Microsoft Internet Explorer and web services. The current release of RMS is Version 1.0SP2.

RMS works as follows. If a user accesses protected content for the first time, she is authenticated through an active directory. If authentication is successful, the user acquires the use license, which is stored locally. This allows working offline in the future.

The module knows three built in rights: Edit, owner and viewrightsdata. The edit right allows a user to decrypt and re encrypt the object. A user that has the owner right is granted to exercise any right on the object. The viewrightsdata right, allows a subject to decrypt the object. Customized rights can be defined. If RMS encounters a self defined right it decrypts the content and passes it to the application. What the user is permitted to do with the decrypted object must be controlled by the application itself.

Further information can be found in [18, 11, 15, 48, 41, 16].

5.2.1 Applicability

Usage Requirements

Usage

RMS supports the usage classes processing, rendering and distribution.

Management is not supported, the encrypted files can be copied and moved without limitation.

Conditions

RMS supports durability, time and cardinality conditions. Action and point in time conditions are not supported. Further RMS allows the revocation of rights, which is an event-defined condition. Revocation needs the user to connect to the RMS server and check if any change of license occurred. The connection can be enforced, by specifying in the license that the newest revocation list must be fetched in regularly intervals.

Enforcement

Use licenses may restrict usage by inhibiting actions or by modifying the effects of rendering if any of the conditions fails. As proposed in [41] a low resolution sample images may be provided. Unless a fee is paid, RMS will replace the action of displaying the original by the action of displaying the sample.

Limitations of Enforcement

Control over distribution is limited. Currently only sending a document by email can be inhibited. However, the encrypted files can be distributed by ftp or the like.

Once content is rendered, no more control is possible. RMS does not use watermarking techniques.

Data

RMS supports protection of arbitrary data.

Environment

Technical

RMS only runs on Microsoft operating systems. To create and consume protected content, an RMS server is needed. The newest server platform that supports RMS is Windows Server 2003. The RMS server needs a database server to store its data. Further it needs an active directory server to authenticate users. Instead of an active directory, Microsoft Passport could be used. Using Passport is discouraged, since the Passport technology provides less security.

RMS clients are available for Windows 98 and newer.

Organizational

The timeliness of the active directory must be assured. Otherwise RMS might not work reliably.

5.2.2 Implementation

Layer of Embedding

RMS is implemented at the operating system layer, since it runs as a privileged kernel module.

Protection Layer

RMS protects objects by encrypting them.

Model

RMS is based on a client server architecture. At least when objects are accessed for the first time, an online connection is needed to get the use license. Therefore RMS is partially online.

Feedback Support

RMS does not support feedback. Feedback mechanisms could be implemented in RMS enabled applications.

Logging Capabilities

RMS installs a logging listener, which runs on both root certification servers and licensing servers. RMS enabled web services log all requests and responses.

The RMS server keeps a log of every license request. It stores the requesters user information, a list of users and groups that possess rights on the relevant object and a list of the rights that are granted to the requester.

License Management

Note that RMS distinguishes publishing and use licenses. A detailed explanation can be found in [10]. We consider use licenses here.

Storage Model

The use license is stored remotely and must be fetched when accessing the content for the first time. After the license is fetched, it is stored with the content.

Negotiation

There is no support for negotiation. Usage rights are automatically provided by the content provider.

Target of License

The license targets subjects in the first. It may also target devices using the lockbox technology. This technology calculates for every device a unique device identifier.

Group Support

Group support is provided by templates. Those allow defining rights for a group of subjects.

Default Behavior

RMS uses the blacklist approach. By default, a document is not protected unless specified.

Rights Expression Language

RMS uses XrML. How XrML rights are interpreted is specified in detail in [41].

5.2.3 Non-functional Properties

Performance

There is a discussion going on about performance implications by DRM systems at the operating system level. Clear is, that the performance is highly dependent on the number of controlled operations. Every access to an RMS protected object must be checked. The more accesses and the more controlled actions the worse performance.

Reliability

The service RMS provides is reliable. Since it supports working offline, there is no need for a permanent connection. However, if one is not able to connect until the license expires the object is no longer accessible.

Usability

Usability for the data consumer as well as the data provider is intuitive. The support for templates allows further simplifications in the process of generating RMS protected objects. Creating and distributing RMS protected content is straight forward and easily learnable.

Costs

Direct Costs

Current licensing fees can be found at [17]. Those are considered medium since RMS addresses corporations and not private persons.

Implementation Expenditure

The implementation expenditure is medium for the distributor and low for the consumer. Most corporates for which RMS is interesting already possess the needed infrastructure.

Installation of the RMS components and some configuration will enable the use of RMS. Because of its infrastructural requirements, RMS is not suitable for use outside of corporate networks.

Security

The security of RMS is medium. It integrates into existing security solutions, uses strong cryptography and does two way authentication of subjects. The design as an operating system module provides better security than a user process could provide.

Controllability

The design as an operating system module and the use of XrML allow for medium controllability.

5.3 Adobe Document Services

"Adobe Document Services" is a solution to manage documents and processes. It consists of a wide range of products. Considering control mechanisms, the core component to look at is the Adobe LiveCycle Policy Server, which offers a multi platform solution for policy management. It protects PDF documents using the J2EE-based server software, widely distributed in Adobe Reader 7.0. Contrary to Microsoft Rights management system, this allows easy distribution of protected documents beyond enterprise. Since Adobe has acquired the FileLine Digital Rights Management division of Navisware, LiveCycle Policy Server supports protection of Microsoft Office documents and CAD formats.

Further information can be found in [45, 4, 5, 47, 3, 2, 1].

5.4 FairPlay

FairPlay is the control mechanism used in Apple iTunes, iPod and QuickTime. FairPlay provides a fixed set of conditions. Protected files may be copied to any number of portable music players. They may also be used on five authorized computers simultaneously. To authorize a computer, iTunes sends a unique machine identifier to Apples servers. In return the computer is provided with all of the users keys needed to unlock the protected files. Deauthorizing a computer works the other way round.

Protected files may be copied on audio CDs without limitation. Therefore the protection that FairPlay provides is very limited. Since neither the audio tracks are protected nor the quality is reduced, it is easy to re-encode the CD and generate unprotected copies of the music files.

Playlists composed in iTunes may be copied to CDs only seven times. Then the playlist must be changed, before new audio CDs can be created.

More information can be found in [27, 36, 40].

5.5 Open Mobile Alliance Digital Rights Management (OMA DRM)

The "Open Mobile Alliance (OMA)" develops specifications for mobile data services. Its goal is to ensure interoperability between multiple devices, providers and markets and to establish a worldwide standard. Intel, mmO2, Nokia, Panasonic, RealNetworks, Samsung and Warner Bros participate in the "Content Management License Administrator". Their goal is to build a business framework for the OMA specification.

Among the many OMA specifications is the OMA DRM specification. OMA DRM aims at protecting mobile content. It can be applied to audio, video and wireless applications. Users should be able to pay once for content and then consume it on any registered mobile device.

Two key technologies that make OMA DRM very interesting are "forward lock", a simple mechanism that prevents content from leaving the phone, and "combined delivery" which allows to further restrict content when redistributing it. The usage rights normally are bound to the content. A technology called "separate delivery" enables to distribute content and usage rights separately. This allows for superdistribution models such as try before buy.

Further information can be found in [38, 8, 7, 9, 6].

5.6 CSS – Content Scramble System

The "Content Scramble System (CSS)" is a protection scheme that encrypts DVD videos. Licensed players contain one or more keys, similar to AACS. To playback a DVD, a licensed player decrypts a key stored on the disc with its own device keys. The disc key is used to decrypt the title key. The title key in turn is used to decrypt the sector key. Finally, the sector key can be used to decrypt the MPEG-2 compressed video data.

CSS does not support any other usage requirements than inhibiting playback on unlicensed devices. Due to fatal design faults, CSS is completely broken.

Further information can be found at [29, 35].

5.7 Windows Media Digital Rights Management (WM DRM)

The "Windows Media Digital Rights Management (WM DRM)" system is aimed at prohibiting pirating of media content. It is embedded in the Windows Media Player. WMV, WMA and ASF files can be protected through encryption. In order to use the protected files, a license needs to be obtained from a server. The license is then stored in a local repository.

Further information can be found at [39, 30, 40].

5.8 HDCP – High-Bandwidth Digital Content Protection

"High-Bandwith Digital Content Protection (HDCP)" allows to control how audiovisual content can be rendered. High-definition digital video can be downgraded to DVD quality on non-HDCP video outputs.

Non-genuine devices are identified by an authentication process. Those devices are only provided with a lower quality version of the content. Data sent over DVI or HDMI interfaces are encrypted. This prevents eavesdropping and "man in the middle" attacks. It is therefore not possible to directly record the video or audio signal.

Key revocation procedures ensure that devices manufactured by vendors who violate the license agreement can be blocked.

HDCP is used in AACS discussed in section 5.1.

Further information can be found in [32, 31, 25, 14, 37]

5.9 DRM/everywhere available (DReaM)

"DRM/everywhere available (DReaM)" is an open source DRM system developed by the OpenMedia Commons. The OpenMedia Commons is an open source community created by Sun Mi-

crosystems. The DReaM architecture is independent of hardware, data format, application domain and operating system. The source code is published under the Common Development and Distribution License (CDDL). There are no licensing fees.

The Sun Labs recently presented two draft specifications called "DReaM-Conditional Access System (DReaM-CAS)" and "DReaM-Mother May I (DReaM-MMI)". DReaM-CAS protects MPEG-2 data streams on IP networks. A prototype implementation of DReaM-CAS is provided by sun. The source code can be downloaded at [42].

DReaM-MMI provides a rights management system for clients directly or indirectly connected to content networks. The key idea is, that a client should have the ability to negotiate for the content of a license. The specification defines the message protocol, message transport and a list of profiles that enable negotiation for rights.

DReaM is based on DRM-OPERA. DRM-OPERA is part of Project OPERA which originated from the Eurescom R&D initiative. It tries to achieve interoperability among different DRM systems. This is done by reducing the complex statements of a license to a lowest common denominator. It only authenticates users and provides one single right: "Play once". The effect is, that many licenses can not be expressed.

Unfortunately, even though the specifications for DReaM-CAS and DReaM-MMI were requested no answer was received. Therefore only limited information from various sources were available.

Further information can be found at [19, 13, 43, 46, 20].

5.10 Content Protection Architecture (CPSA)

The "Content Protection Architecture (CPSA)" is an architecture mainly developed by Intel, IBM, Matsushita und Toshiba. It consists of the "Content Protection for Prerecorded Media (CPPM)" and the "Content Protection for Recordable Media (CPRM)".

CPPM is an extension of CSS discussed in section 5.6. Contrary to CSS there exists only an album identifier and a "Media Key Block" that is used for the decryption of content.

CPPM supports DVD media, IBM-Microdrive harddiscs, SD memory cards and Compact-Flash cards. It is fully compatible with CSS. No special player is needed to play CPPM protected DVDs.

CPRM allows to record protected content. It describes how data has to be stored on rerecordable media.

Further information can be found in [28].

6 Outlook and Conclusion

6.1 Current Status

A wide variety of products exists. Many of the systems that are available today are proprietary and self-contained systems. They are not interoperable and no standards exist. The development is mainly driven by the music and movie industry. Therefore most control mechanisms are tailored toward this sector. Although sophistic mechanisms are designed, development is hindered by patents. There exist entire companies, that just collect patents and collect fees.

Most of the mechanisms analyzed in this report control the usage classes processing and rendering. Management and distribution are rarely covered. Designers argue that the objects are encrypted and copying or distributing them does not cause any harm. While this might be true in today's application domains of DRM, in the future it might not. The cryptography used today

might be easily breakable in some years. Also there are usage scenarios where a content distributor might desire to use short keys for enhanced performance in combination with time conditions. It then must be ensured that objects do not exist longer than intended or the key could be bruteforced.

Time and cardinality conditions are covered by almost any of the more sophisticated mechanisms. Time conditions are mostly limited to durability. The only event-defined conditions deal with revocation and license changes. Purpose of use and governance restriction are not supported by any of the mechanisms. This is comprehensible, since such conditions are hard or even impossible to enforce.

Control mechanisms are usually implemented at the application layer or at the hardware layer. The mechanisms implemented at the hardware layer are basically used in specialized devices, for example video players. RMS is the only control mechanism that is implemented at the operating system layer.

Most designers of control mechanisms do not provide detailed information about logging and feedback capabilities. Often they are not even mentioned. This most probably is because people generally are skeptical of DRM and will refuse to use mechanisms that implement logging or feedback.

Other than Adobe, manufacturers of proprietary mechanisms do not name which rights expression language is used. Further, most usage requirements that the mechanism supports are provided with examples, which makes it somewhat cumbersome to collect all the information needed for classification.

6.2 Trends

There is a strong need for interoperability and open standards. This need arouse from the fact that many of today's devices support sharing media. A movie may be played on a PC, a TV that gets the content from a media server or even on mobile phones. Content providers must provide the content in such a way, that the user is still able to use it on arbitrary devices. Otherwise consumers will not buy it. Patents make it difficult to come up with open specifications and implementations. There is a trend to circumvent those restrictions by trying to invent new, unpatented techniques and releasing them under an open license.

The improved support for a variety of use cases makes control mechanisms interesting for business. Control mechanisms will be integrated into business processes to protect company assets. This will allow companies to avert leakage of confidential information. Further, a company may provide its customer with confidential information and must not fear that the customer will release it to third parties.

The layer of embedding shifts more and more toward the hardware layer. At least parts of the control mechanisms, such as keys or random number generators, will be embedded in hardware in most of the coming mechanisms. Control mechanisms at the hardware layer are more robust and allow for better controllability, because a single interface is provided to the operating system and the applications. Further, the use of analog outputs will be disencouraged. Digital outputs allow protection just before content is made audible or visible. This makes it hard to rerecord high quality content.

Control mechanisms more and more are modularly designed. This allows to combine certain modules that may collaborate to cover the desired usage requirements as exact as possible.

6.3 Future Work

This work revealed, that control mechanisms are rarely used in privacy context. Based on this work, appropriate mechanisms could be chosen and adopted for use in privacy context. Especially the open systems provide a good basis for this task.

To improve this taxonomy, one could work out a structure that allows quick decision which mechanisms can be combined. While combination is not that important today – as today's systems are seldom interoperable – this will become important in the future.

Further, formal criteria could be defined. Together with a formal description of the protection requirements, optimal mechanisms can be chosen. The optimality of a mechanisms could be proved. Because of the many aspects and facets in the field of control mechanism, this task appears to be very difficult.

Having defined formal criteria, a program that automatically selects a control mechanism which meets certain use cases could be designed. Automated selection of control mechanisms could also be implemented in software design tools. This would speed up and improve the security of system development.

6.4 Experiences

To dive into the huge field of DRM and control mechanism seems easy in the first place. Lots of resources can be found. There are many good papers about how DRM works generally. However, when looking for specific information about proprietary mechanisms the affair becomes difficult. Most of the vendors do not hesitate to provide examples about what their system can do and what cryptographic techniques are applied, but they do not provide accurate statements about what usage requirements are supported. On the other hand, open systems are usually very well documented. This is comprehensible, since they try to gain wide acceptance.

Since it was the goal to analyze very different systems in the context of usage control, it was indispensable to come up with an appropriate definition of control mechanism. A further complication was that usage control is a very young field. Most of the insights about usage, conditions and enforcement were only developed during the work. Things had to be changed often, which had further implications on the classification criteria. It was very interesting to see a huge pool of ideas and presumably intuitive understanding converge toward well defined concepts.

During the research, several other surveys about DRM mechanisms were found. However, those were limited to systems with similar properties. For example, WM DRM, FairPlay and the like were compared. Others compared copy protection schemes. But none of them were constructed to support the comparison of arbitrary systems as it is done in this work.

This work allowed to gain an overview over many different techniques. Assets and drawbacks of usage control were discovered. The knowledge gained during this work allows for neutral judgment of techniques and encourages to participate in the research and development concerned with usage control.

A Taxonomy Overview

Symbols are explained in Table 1.

Applicability	AACS	RMS	Adobe Document Services
Usage Requirements			
Usage	Processing, Rendering, Management, Distribution	Processing, Rendering, Distribution	Processing, Rendering
Conditions			
Action	◐	◐	◐
Time	◐	◐	●
Cardinality	◐	●	○
Event-defined	◐	◐	◐
Purpose of Use	○	○	○
Governance	○	○	○
Technical	○	○	○
Enforcement	Inhibit, Modify/Replace	Inhibit, Modify/Replace	Inhibit, Modify/Replace
Limitations of Enforcement	Limited control after rendering through support for audio watermarks. Media might be copied on physical layer.	No control after Rendering. Encrypted objects may be distributed without control.	Limited control after rendering through support for watermarks.
Data	Audiovisual	Arbitrary Data	Adobe PDF documents, MS Office + CAD documents.
Environment			
Technical	AACS compliant PC/CE devices, AACS compliant Optical media	RMS Server, Active Directory or MS Passport, Database Server, MS Operating System	Adobe LiveCycle environment, Adobe LiveCycle policy server
Organizational	Licensing entity, Trusted replicator	Ensure up to date Active Directory	✗

Applicability	FairPlay	OMA DRM 2.0	CSS
Usage Requirements			
Usage	Processing, Rendering	Processing, Rendering, Distribution	Processing, Rendering
.....	.....	.....	.....
Conditions			
Action	◐	◐	○
Time	○	●	○
Cardinality	◐	●	○
Event-defined	○	◐	○
Purpose of Use	○	○	○
Governance	○	○	○
Technical	◐	◐	◐
Enforcement	Inhibit	Inhibit Modify/Replace	Inhibit
Limitations of Enforcement	No control after Rendering. No control after burning audio CD.	Release to selected systems in well defined manner.	Completely Broken.
Data	AAC Audio MPEG-4 and H.264 video	Arbitrary	DVD-Movies
Environment			
Technical	iTunes, iPod or QuickTime	OMA DRM compliant device	CCA authorized Device
.....	.....	.....	.....
Organizational	✗	✗	✗

Applicability	WMDRM	HDCP	DReaM
Usage Requirements			
Usage	Processing, Rendering	Rendering, Management	Processing, Rendering
Conditions			
Action	◐	○	○
Time	●	○	○
Cardinality	◐	○	●
Event-defined	◐	◐	○
Purpose of Use	○	○	○
Governance	○	○	○
Technical	○	●	○
Enforcement	Inhibit	Inhibit, Modify/Replace	Inhibit
Limitations of Enforcement	No protection after Rendering.	No protection after Rendering. Encrypted content may be recorded at digital outputs.	No protection after Rendering.
Data	ASF, WMV and WMA	Audiovisual	Arbitrary
Environment			
Technical	MS Windows PC or portable device	HDCP compliant device	✗
Organizational	✗	✗	✗

Applicability	CPSA		
Usage Requirements			
Usage	Processing, Rendering		
.....	.....	.....	.....
Conditions			
Action	○		
Time	○		
Cardinality	●		
Event-defined	◐		
Purpose of Use	○		
Governance	○		
Technical	●		
Enforcement	Inhibit		
Limitations of Enforcement	No control after Rendering.		
Data	Audiovisual content		
Environment			
Technical	DVD-Media, IBM-Microdrive-Harddisc, SD-Memory-Card or Compact-Flash-Card		
.....	.....	.....	.....
Organizational	✗		

Implementation	AACS	RMS	Adobe Document Services
Layer of Embedding	Hardware layer, Software layer (optionally)	Operating system layer	Software layer, Hardware + Software layer (optionally)
Protection Layer	Data layer, Device layer	Data layer	Data layer
Model	Local, Client-Server (Optionally, partially online)	Client-Server (Partially online)	Local, Client-Server (License dependent, partially online)
Feedback Support	✓	✗	✓
Logging Capabilities	✗	✓	✓
License			
Storage Model	With content	Remote repository, With content	With content
Negotiation	Automated, data provider	Automated, data provider	Automated, data provider
Target of License	Device	Subject, Device	Subject
Group Support	✓	✓	✓
Default Behavior	Whitelist	Blacklist	Fixed
Rights Expression Language	?	XrML	PDRL

Implementation	FairPlay	OMA	CSS
Layer of Embedding	Application layer	Application layer	Application layer, Hardware layer
Protection Layer	Data layer	Data layer	Data layer
Model	Local	Local	Local
Feedback Support	✓	?	✗
Logging Capabilities	✗	?	✗
License			
Storage Model	With content	Local repository	With content
Negotiation	Automated, data provider	Automated, data provider	Automated, data provider
Target of License	Device	Device, Subject	Device
Group Support	✗	✓	✓
Default Behavior	Fixed	Whitelist	Fixed
Rights Expression Language	✗	ODRL	✗

Implementation	WMDRM	HDCP	DReaM
Layer of Embedding	Application layer	Hardware layer	Application layer
Protection Layer	Data layer	Data layer	Data layer
Model	Client-Server (Partially online)	Client-Server (Online)	Client-Server (Partially online)
Feedback Support	✓	✗	?
Logging Capabilities	✗	✗	?
License			
Storage Model	Remote repository, Local repository	With content	Remote repository, Local repository
Negotiation	Automated, data provider	Automated, data provider	Automated, bidirectional
Target of License	Device	Device	Subject
Group Support	✗	✓	?
Default Behavior	?	?	?
Rights Expression Language	?	?	?

Implementation	CPSA		
Layer of Embedding	Hardware Layer, Application layer		
Protection Layer	Data layer		
Model	Local		
Feedback Support	✗		
Logging Capabilities	✗		
License			
Storage Model	With content		
Negotiation	Automated, data provider		
Target of License	Device		
Group Support	✓		
Default Behavior	Fixed		
Rights Expression Language	✗		

Non-functional Properties	AACS	RMS	Adobe Document Services
Performance	✭✭✭ (Constant)	✭✭ (Variable)	✭✭✭ (Constant)
Reliability	✭✭✭	✭✭	✭✭✭
Usability			
Consumer:	✭✭✭	✭✭✭	✭✭✭
Distributor:	✭✭✭	✭✭✭	✭✭✭
Costs			
Direct Costs			
Consumer:	✭✭✭	✭	✭
Distributor:	✭✭✭	✭✭	✭✭
.....	.....	.....	.....
Implementation Expenditure			
Consumer:	✭✭✭	✭	✭
Distributor:	✭✭✭	✭✭	✭
Security	✭✭✭	✭✭	✭✭
Controllability	✭✭✭	✭✭	✭✭

Non-functional Properties	FairPlay	OMA	CSS
Performance	✭✭✭ (Constant)	? (?)	✭✭✭ (Constant)
Reliability	✭	✭✭✭	✭
Usability			
Consumer:	✭✭✭	✭✭✭	✭
Distributor:	✭✭✭	✭✭	✭✭✭
Costs			
Direct Costs			
Consumer:	✭✭	?	✭
Distributor:	✭	?	✭
.....	.....	.....	.....
Implementation Expenditure			
Consumer:	✭	✭✭	✭
Distributor:	✭✭	✭✭	✭✭
Security	✭	✭✭✭	✭
Degree of Controllability	✭	✭✭✭	✭

Non-functional Properties	WMDRM	HDCP	DReaM
Performance	✭✭✭ (Constant)	✭✭✭ (Constant)	✭✭ (Constant)
Reliability	✭✭	✭✭✭	✭✭
Usability			
Consumer:	✭✭✭	✭✭✭	✭
Distributor:	✭✭✭	✭✭	✭✭
Costs			
Direct Costs			
Consumer:	✭	✭✭✭	✭
Distributor:	?	?	✭
.....	.....	.....	.....
Implementation Expenditure			
Consumer:	✭	✭✭✭	✭
Distributor:	✭	?	✭✭✭
Security	✭✭	✭✭✭	✭✭
Degree of Controllability	✭✭	✭✭✭	✭✭

Non-functional Properties	CPSA		
Performance	✭✭✭ (Constant)		
Reliability	✭✭✭		
Usability			
Consumer:	✭✭✭		
Distributor:	✭✭		
Costs			
Direct Costs			
Consumer:	✭✭		
Distributor:	?		
.....	.....	.....	.....
Implementation Expenditure			
Consumer:	✭✭		
Distributor:	✭✭		
Security	✭✭		
Degree of Controllability	✭✭		

✓	Available	✯	Low / Complicated
✗	Not available / None	✯✯	Medium / Learnable
○	Not supported	✯✯✯	High / Intuitive
◐	Partially supported	A + B	A combined with B
●	Fully supported	?	Unknown

Table 1: Symbols used for classification.

Contents

List of Figures

List of Tables

References

[1] Adobe. Adobe livecycle document security. http://www.adobe.com/products/server/securityserver/pdfs/docsecurityserver_ds.pdf.

[2] Adobe. Adobe livecycle policy server. http://www.adobe.com/products/server/policy/index.html.

[3] Adobe. Document control and security. http://www.adobe.com/security/index.html.

[4] Adobe. A primer on electronic document security. http://www.adobe.com/security/pdfs/acrobat_security_wp.pdf.

[5] Adobe. Portable document rights language (pdrl) specification, version 7.0. http://www.adobe.com/devnet/livecycle/policyserver/articles/pdrl.pdf, 2005.

[6] Open Mobile Alliance. Open digital rights language (odrl), version 1.1. http://odrl.net/1.1/ODRL-1.1.pdf, August 2002.

[7] Open Mobile Alliance. Drm architecture, approved version 2.0, March 2006.

[8] Open Mobile Alliance. Drm specification, approved version 2.0, March 2006.

[9] Open Mobile Alliance. Oma drm requirements, approved version 2.0, March 2006.

[10] Starr Andersen. Rms technical reference, April 2005.

[11] Urs Bertschy. Windows rms: Mehr schutz für dokumente. http://www.infoweek.ch/archive/ar_single.cfm?ar_id=13356&ar_subid=2&sid=0.

[12] CDRinf. Aacs content protection scheme details. http://www.cdrinfo.com/Sections/News/Details.aspx?NewsId=17298.

[13] Open Media Commons. Trust is the currency of the participation age. http://www.openmediacommons.org/.

[14] Digital Connection. What is hdcp? http://www.digitalconnection.com/FAQ/HDTV_12.asp.

[15] Mircrosoft Corporation. Windows rights management services. http://www.microsoft.com/windowsserver2003/technologies/rightsmgmt/default.mspx.

[16] Mircrosoft Corporation. Windows rights management services. http://blogs.msdn.com/rms.

[17] Mircrosoft Corporation. Windows rights management services for windows server 2003 pricing and licensing overview. http://www.microsoft.com/windowsserver2003/techinfo/overview/rmsoverview.mspx#EVE.

[18] Ernst and Young. Overview of microsoft rights managment services. http://www.ey.com/global/download.nsf/US/Microsoft_Management_Services/$file/MicrosoftRMS.pdf.

[19] Gerard Fernando, Tom Jacobs, and Vishy Swaminathan. Project dream an architectural overview. http://www.openmediacommons.org/collateral/DReaM-Overview.pdf, September 2005.

[20] heise online. Quelloffene digitale nutzungskontrolle von sun. http://www.heise.de/newsticker/meldung/71221, March 2006.

[21] AACS LA. Advanced access content system licensing administrator. http://www.aacsla.com.

[22] AACS LA. Advanced access content system (aacs), introduction and common cryptographic elements, revision 0.91, February 2006.

[23] AACS LA. Advanced access content system (aacs), pre-recorded video book, revision 0.91, February 2006.

[24] AACS LA. Advanced access content system (aacs), recordable video book, revision 0.91, February 2006.

[25] THG Lexikon. High-bandwidth digital content protection. http://www.thgweb.de/lexikon/High-bandwidth_Digital_Content_Protection.

[26] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. Digital rights management for content distribution. In *CRPITS '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 49–58, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[27] Apple Computer, Inc. ipod + itunes. http://www.apple.com/itunes.

[28] Diran Al-Dairani, Ljubomir Ovtscharov, Christian Siefkes, Rolf Thomasius. Content management systeme - sinn und unsinn des kopierschutz. http://page.mi.fu-berlin.de/~siefkes/cms/cms.html, July 2001.

[29] DVD Copy Control Association. Content scramble system (css). http://www.dvdcca.org/css/.

[30] Microsoft. Digital rights management (drm). http://www.microsoft.com/windows/windowsmedia/forpros/drm/default.mspx.

[31] Digital Content Protection, LLC. High-bandwidth Digital Content Protection (HDCP). http://www.digital-cp.com/.

[32] Digital Content Protection, LLC. High-bandwidth digital content protection system, revision 1.2. http://www.digital-cp.com/home/HDCPSpecificationRev1_1.pdf, June 2006.

[33] The Guardian. Has hollywood gone overboard on privacy? http://technology.guardian.co.uk/online/insideit/story/0,,1752293,00.html.

[34] Wikipedia, the free encyclopedia. Blu-ray disc. http://en.wikipedia.org/wiki/Blu-ray_Disc.

[35] Wikipedia, the free encyclopedia. Content scrambling system. http://de.wikipedia.org/wiki/Content_Scrambling_System.

[36] Wikipedia, the free encyclopedia. Fairplay. http://en.wikipedia.org/wiki/FairPlay.

[37] Wikipedia, the free encyclopedia. High-bandwidth digital content protection. http://en.wikipedia.org/wiki/High-Bandwidth_Digital_Content_Protection.

[38] Wikipedia, the free encyclopedia. Open mobile alliance. http://de.wikipedia.org/wiki/Open_Mobile_Alliance.

[39] Wikipedia, the free encyclopedia. Windows media drm. http://en.wikipedia.org/wiki/WMDRM.

[40] Sam Michiels, Wouter Joosen, Eddy Truyen, and Kristof Verslype. Digital rights management – a survey of existing technologies, November 2005.

[41] Microsoft. Rights managment services sdk. http://windowssdk.msdn.microsoft.com/en-us/library/ms716148.aspx.

[42] Sun Microsystems. java.net. http://dream.dev.java.net/.

[43] Sun Microsystems. Open media commons releases specifications and source code for Open, royalty-free digital rights management. http://www.sun.com/smi/Press/sunflash/2006-03/sunflash.20060321.2.xml, March 2006.

[44] Alexander Pretschner, Manuel Hilty, and David Basin. Usage control. *Communications of the ACM*, September 2006. To appear.

[45] Gartner RAS Core Research. Navisware e-drm buy could give adobe a one-stop-shopping solution. http://www.adobe.com/manufacturing/pdfs/gartner_1691.pdf, January 2006.

[46] Bill Rosenblatt. Sun's open-source dream. http://www.drmwatch.com/special/article.php/3531651, September 2005.

[47] SAP. Adobe document services. http://help.sap.com/saphelp_nw2004s/helpdata/de/3e/13bcb17fa242279401aa8567bf76d3/content.htm.

[48] Microsoft TechNet. Managing rms. http://technet2.microsoft.com/WindowsServer/en/Library/7eb5cdd1-cd48-4b2b-96b6-fc74f7b42e7f1033.mspx?mfr=true.