Leliwa Technical Bulletin

# OMA DRM

**Date:** 14.12.2009

**Revision:** 002/DRM/009

**Author:** Jakub Bluszcz

LELIWA

# Table of contents

# Introduction

Historically, content such as books, music, games, and videos have been delivered on paper, magnetic tape, and disks. The technology required to digitally copy and redistribute this content on a large scale prohibited the secondary market from having much affect on revenues from content sales. With large decreases in the cost of technology, e.g. storage space and recordable digital media, and greater Internet bandwidth, services like Napster and Gnutella have sprung up to allow massive redistribution of music and similar content. At the same time, the absence of protection of rights associated with this kind of content in the digital environment has so far prevented the use of Internet as a distribution channel for valuable content.

With the advent of faster wireless networks and increasingly capable user equipment, the mobile environment will soon become another avenue for distributing valuable content. This will require taking steps to establish a model for protecting the rights of the content providers when distributing digital content in the mobile environment.

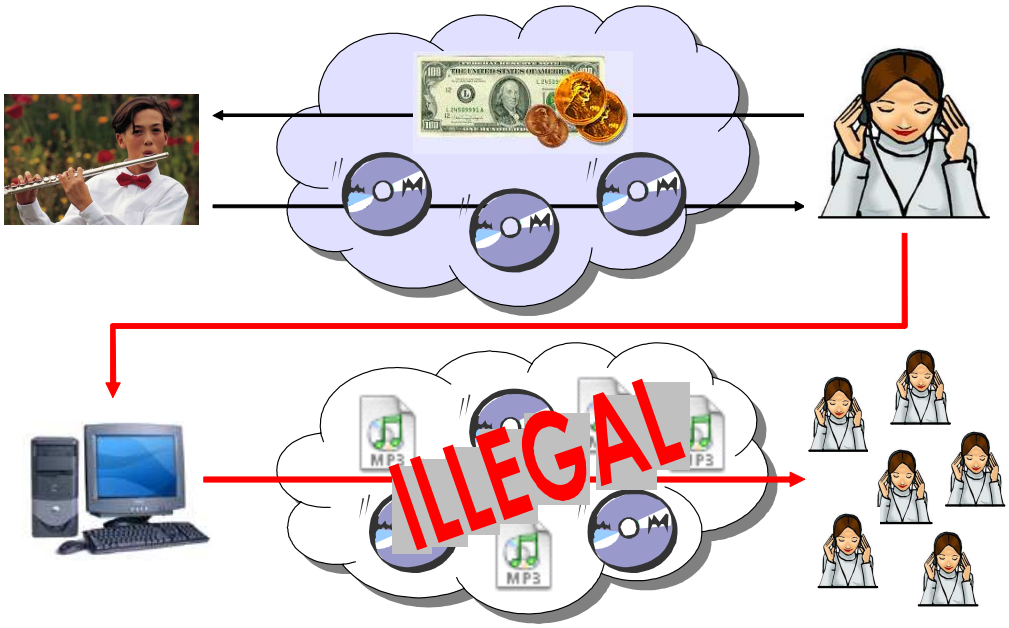


*Figure 1 Distribution without DRM system*

The role of DRM in distribution of content is to enable business models whereby the consumption and use of content is controlled. As such, DRM extends beyond the physical delivery of content into managing the content lifecycle. When a user buys content, she may agree to certain constraints - for example by choosing between a free preview version or a full version at cost, or she may agree to pay a monthly fee. DRM allows this choice to be translated into permissions and constraints, which are then enforced when the user accesses the content.

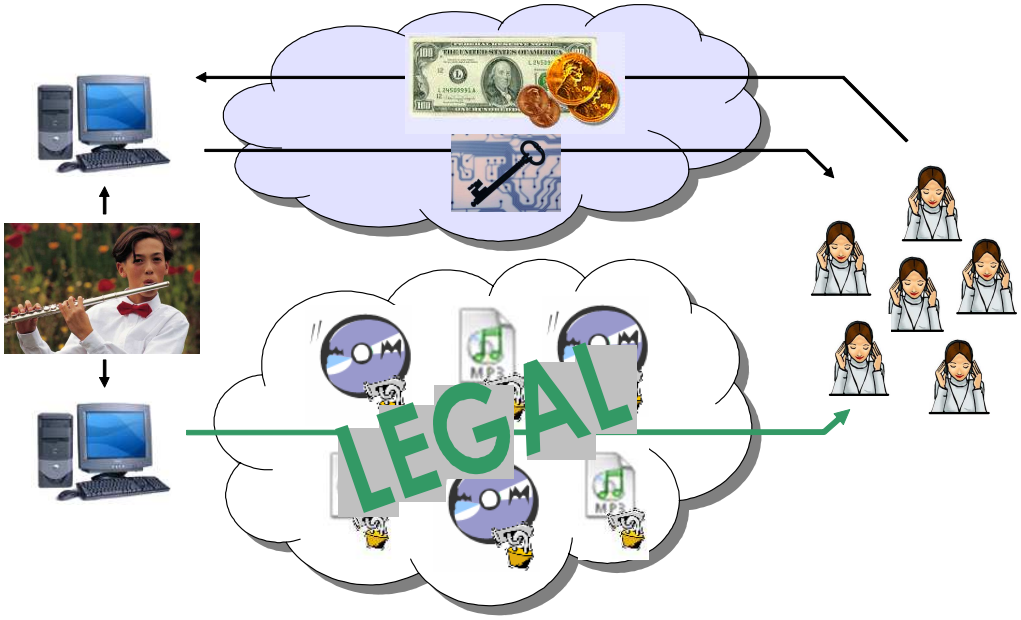


*Figure 2 Distribution with DRM system*

There are different DRM systems. All of them potentially can be deployed in the 3G mobile environment since it provides broadband Internet connectivity. The 3GPP in its early standards has clearly stated that the standard DRM solution will be based on the Open Mobile Alliance (OMA) DRM system. Since OMA DRM solution has a support of both: 3GPP and 3G terminal vendors it is clear that this solution has the greatest chance to be widely deployed across 3G networks worldwide.

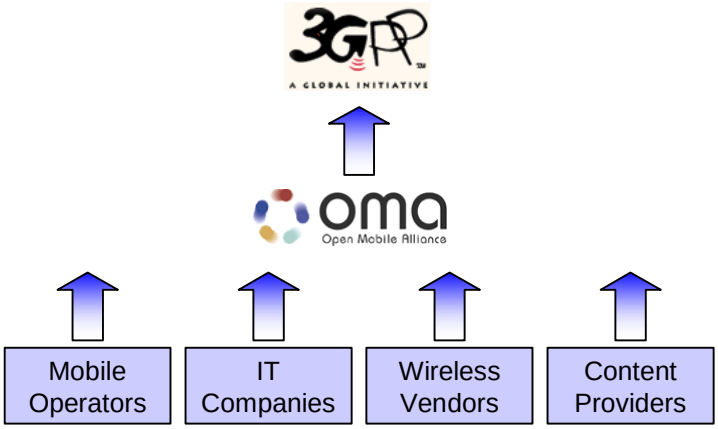


*Figure 3 DRM standardisation bodies*

OMA DRM system specification is released in phases. Majority of new mobile terminals supports either some selected mechanism of the OMA DRM system or even the full set of OMA DRM Phase 1 mechanisms. OMA DRM Phase 2 will be implemented in the future mobile terminals.

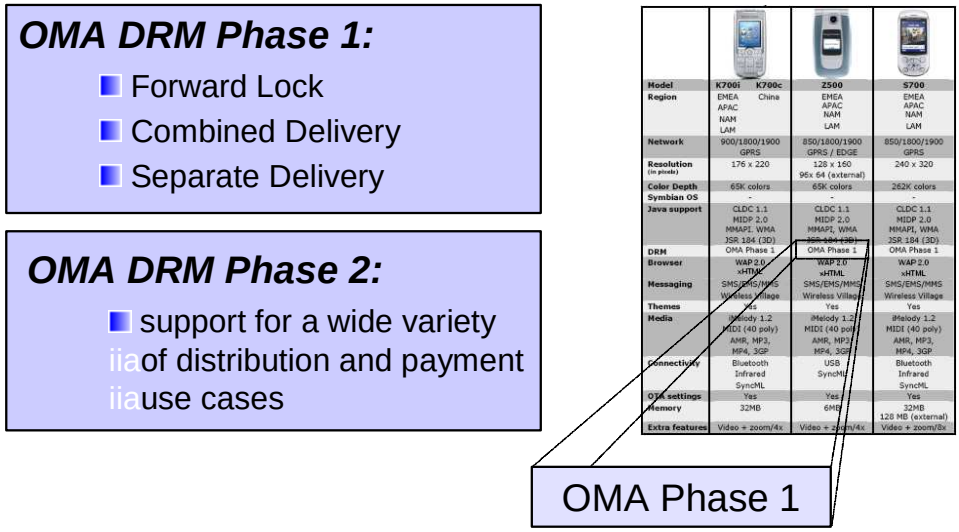


*Figure 4 OMA DRM Phases*

# OMA DRM v1

The OMA DRM v1 Enabler Release was developed rapidly in order to reduce time to market. The Enabler Release was published in November 2002 and was immediately available for companies to implement in their mobile products.

OMA DRM 1.0 includes three levels of functionality:

- Forward Lock - prevents content from leaving device,
- Combined Delivery - adds rights definition,
- Separate Delivery - provides content encryption and supports superdistribution.

## Forward Lock

The purpose of Forward Lock is to prevent peer-to-peer distribution of low-value content. This applies often to subscription-based services, such as news, sports etc. The plaintext content is packaged inside a DRM message that is delivered to the terminal. The device is allowed to play, display or execute the content, but it cannot forward the object.

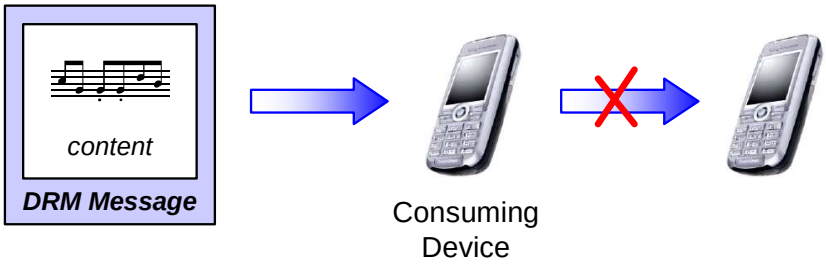


*Figure 5 Forward Lock*

## Combined Delivery

Combined delivery equally prevents peer-to-peer distribution, but it also controls the content usage. In combined delivery method, the DRM message contains two objects, the content and a rights object. The rights object defines permissions and constraints for the use of content. These can be, for example a permission to play a tune only once, or using the content only for x number of days. Neither content nor the rights object can be forwarded from the target device.

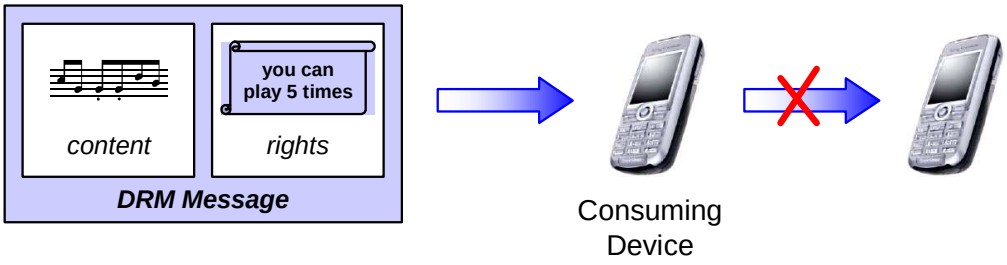


*Figure 6 Combined Delivery*

## Separate Delivery

The purpose of Separate Delivery is to protect higher value content. It enables so called superdistribution, which allows the device to forward the content, but not the usage rights. This is achieved by delivering the media and usage rights via separate channels. The content is encrypted into DRM Content Format (DCF) using symmetric encryption; the DCF provides plaintext headers describing content type, encryption algorithm, and other useful information. Rights object holds the symmetric Content Encryption Key (CEK), which is used by the DRM User Agent in the device for decryption. The Rights Object is created by using OMA Rights Expression Language (REL). OMA Right Expression Language is a mobile profile of ODRL (Open Digital Rights Language) 1.1.

Superdistribution is an application of Separate Delivery that also requires a Rights Refresh mechanism that allows additional rights for the media. Recipients of superdistributed content must contact the content retailer to obtain rights to either preview or purchase the media. Thus, the separate

delivery method enables viral distribution of media maximizing the number of potential customers while retaining control for the content provider through centralised rights acquisition.

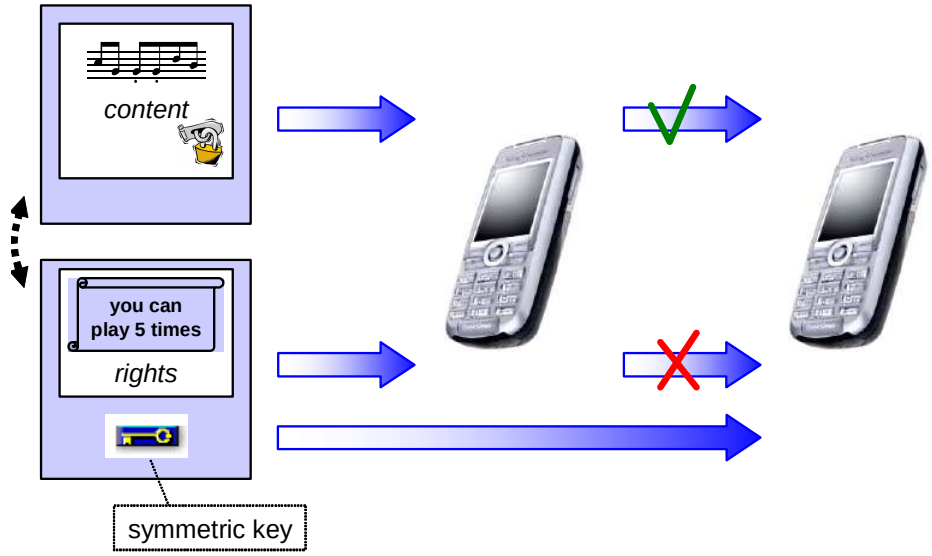


*Figure 7 Separate Delivery*

# OMA DRM v2

DRM solution is evolving with the mobile industry. The higher bandwidth provided by 2,5G and 3G cellular networks allow larger content files to be transmitted over the air. Proliferation of wireless Internet “hotspots” makes Internet access easily available to consumers. Smart mobile devices with removable media and larger colour screens support downloading and streaming rich media content. Content and service providers are eager to release rich audio/video content and applications into the mobile marketplace. All these factors contribute to the requirements of continuously enhanced OMA DRM solution. Greater security and trust management is required to protect the high value content. There’s a need to ensure that the target device can be trusted to keep the content and secrets safe. Greater security is also needed in order to prevent content from leaking out during the acts of downloading and other distribution. The Open Mobile Alliance is meeting these market needs by upgrading the existing OMA DRM Enabler Release with enhanced features. The next version of OMA DRM adds enhanced security by encrypting the rights object and the content encryption key by using the device’s public key to bind them to the target device. Integrity protection for both content and the rights object will be added to reduce the risk of tampering. In addition to these enhanced security features, additional trust elements will be introduced. Mutual authentication between the device and the rights issuer, i.e. the content retailer, will add trust to the

downloading or messaging scenario. The rights issuer will be able to accurately identify the device in order to determine the revocation status of the transaction.

New version of OMA DRM will also support for a wide variety of distribution and payment use cases.

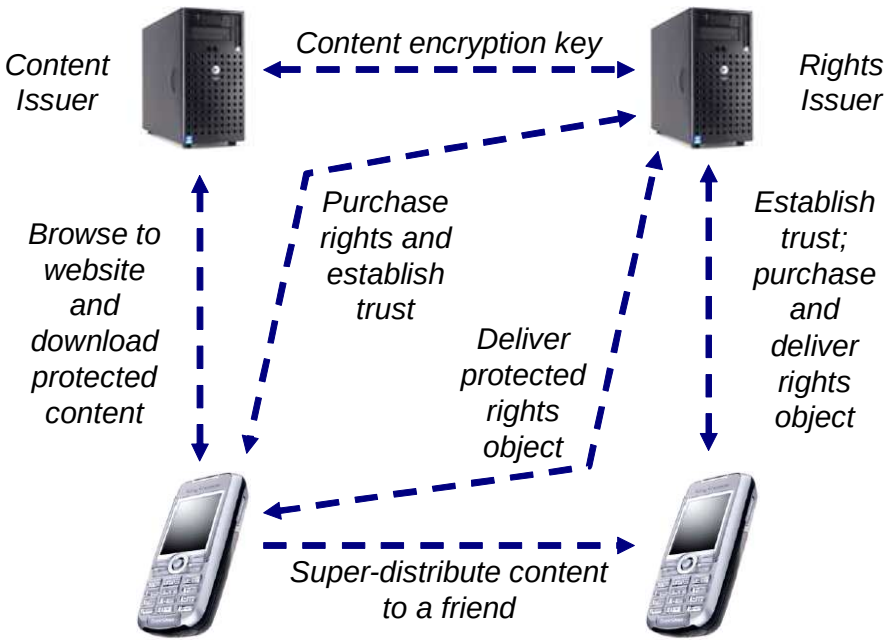


*Figure 8 Example of DRM employment*

# Actors and Functional Entities

In the OMA DRM architecture, functional entities are used to embody specific roles in the DRM system. This makes it possible to decompose the tasks involved in digital rights management, separately from what actors perform each task in a certain deployment.

The functional entities are logical and need not represent physical network nodes (servers, etc). Depending on configuration, different functional entities may be implemented by the same or different physical nodes, and be operated by the same or different actors. Different deployments may incorporate some or all of the functional entities depending on the required functionality in each deployment setting.

From the point of view of digital rights management, the following functional entities have been identified in the architecture:

### *DRM Agent*

A DRM Agent embodies a trusted entity in a device. This trusted entity is responsible for enforcing permissions and constraints associated with DRM Content, controlling access to DRM Content, etc.

*Figure 9 DRM Agent*

### *Content Issuer*

The content issuer is an entity that delivers DRM Content. DRM defines the format of DRM. The content issuer may do the actual packaging of DRM Content itself, or it may receive pre-packaged content from some other source.

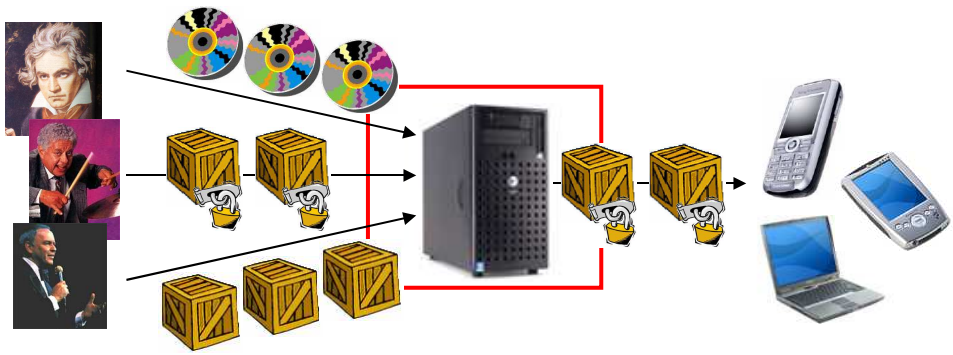

*Figure 10 Content issuer*

Content delivered to DRM Agents, and the way DRM Content can be transported from a content issuer to a DRM Agent using different transport mechanisms.

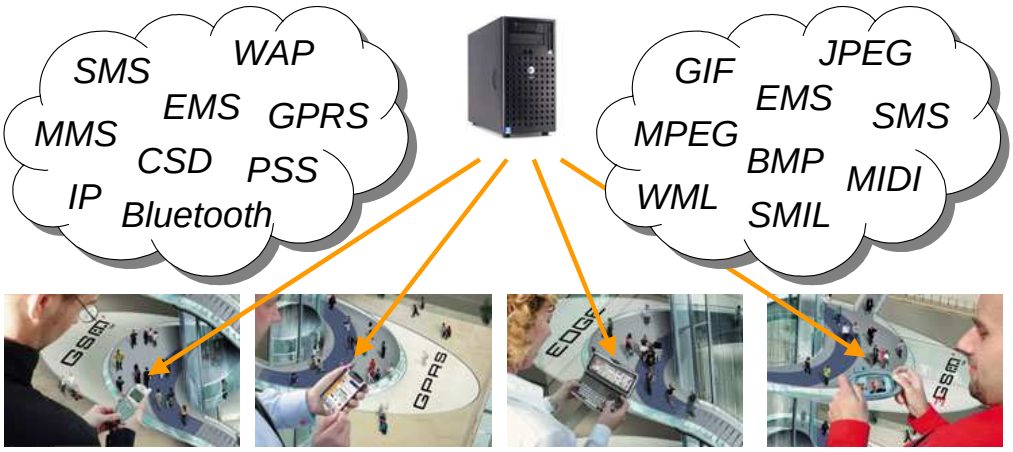


*Figure 11 Transport mechanisms*

### *Rights Issuer*

The rights issuer is an entity that assigns permissions and constraints to DRM Content, and generates Rights Objects. A Rights Object is an XML document expressing permissions and constraints associated with a piece of DRM Content. Rights Objects govern how DRM Content may be used – DRM Content cannot be used without an associated Rights Object, and may only be used as specified by the Rights Object.

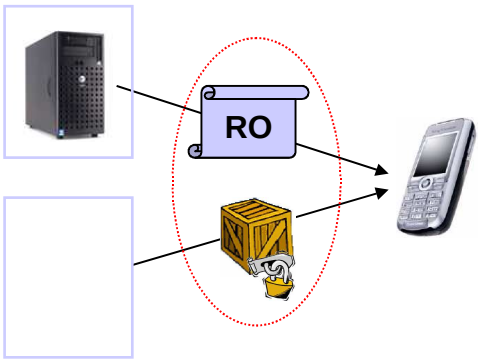


*Figure 12 Rights issuer*

### *User*

A user is the human user of DRM Content. Users can only access DRM Content through a DRM Agent.


A user is the human user of DRM Content. Users can only access DRM Content through a DRM Agent.

*Figure 13 User*

### *Off-device Storage*

DRM Content is inherently secure, and may be stored by users off-device - for example in a network store, a PC, on removable media or similar. This may be used for backup purposes, to free up memory in a device, and so on. Similarly, Rights Objects that only contain stateless permissions may be stored off-device.

*Figure 14 Off-device Storage*

# Functional Architecture

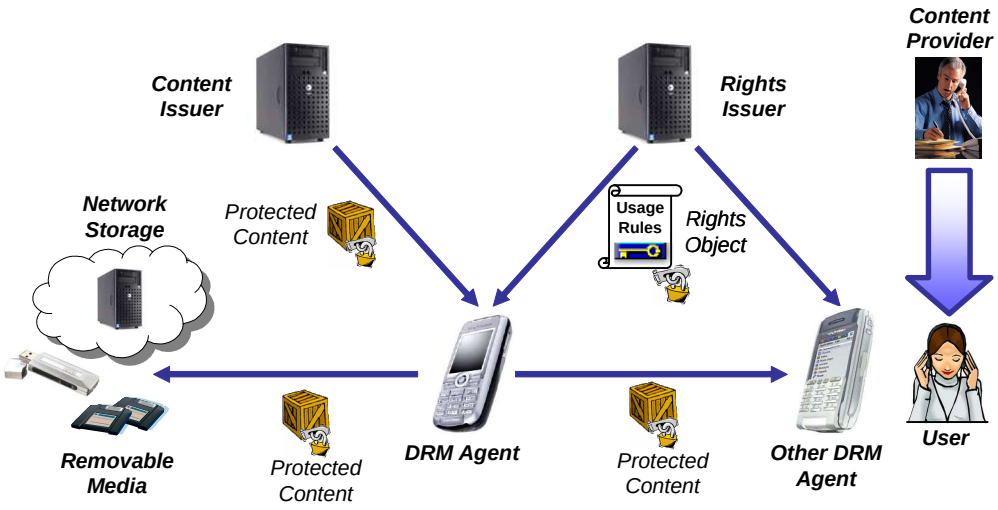


*Figure 15 Functional Architecture*

Before content is delivered, it is packaged to protect it from unauthorised access. A content issuer delivers DRM Content, and a rights issuer generates a Rights Object. The content issuer and rights issuer embody roles in the system. Depending on deployment they may be provided by the same or different actors, and implemented by the same or different network nodes. For example, in one deployment, content owners may pre-package DRM Content, which is then distributed by a content distributor acting as both content issuer and rights issuer.

A Rights Object governs how DRM Content may be used. It is an XML document specifying permissions and constraints associated with a piece of DRM Content. DRM Content cannot be used without an associated Rights Object, and may only be used according to the permissions and constraints specified in a Rights Object.

OMA DRM makes a logical separation of DRM Content from Rights Objects. DRM Content and Rights Objects may be requested separately or together, and they may be delivered separately or at the same time. For example, a user can select a piece of content, pay for it, and receive DRM Content and a Rights Object in the same transaction. Later, if the Rights Object expires, the user can go back and acquire a new Rights Object, without having to download the DRM Content again.

Rights Objects associated with DRM Content have to be enforced at the point of consumption. This is modelled in the OMA DRM specifications by the introduction of a DRM Agent. The DRM Agent embodies a trusted component of a device, responsible for enforcing permissions and constraints for DRM Content on the device, controlling access to DRM Content on the device, and so on.

A Rights Object is cryptographically bound to a specific DRM Agent, so only that DRM Agent can access it. DRM Content can only be accessed with a valid Rights Object, and so can be freely distributed. This enables, for example, superdistribution, as users can freely pass DRM Content between them. To access DRM Content on the new device, a new Rights Object has to be requested and delivered to a DRM Agent on that device.

If rights issuers support it, a Rights Object may optionally be bound to a group of DRM Agents. This is known in the OMA DRM specifications as a Domain. DRM Content and Rights Objects distributed to a domain can be shared and accessed off-line on all DRM Agents belonging to that domain. For example, a user may purchase DRM Content for use on both her phone and her PDA.

The OMA DRM specifications define the format and the protection mechanism for DRM Content, the format (expression language) and the protection mechanism for the Rights Object, and the security model for management of encryption keys. The OMA DRM specifications also define how DRM Content and Rights Objects may be transported to devices using a range of transport mechanisms, including pull (HTTP Pull, OMA Download), push (WAP Push, MMS) and streaming.

# DRM Content Format

Within OMA DRM, Media Objects are encrypted and packaged into a specific format, the DRM Content Format (DCF). The DCF can be delivered separately from an associated Rights Object, which contains the encryption key used to encrypt the Media Object.

The DRM Content Format is closely related to the Rights Expression Language, which defines the syntax and semantics for the Rights Objects.

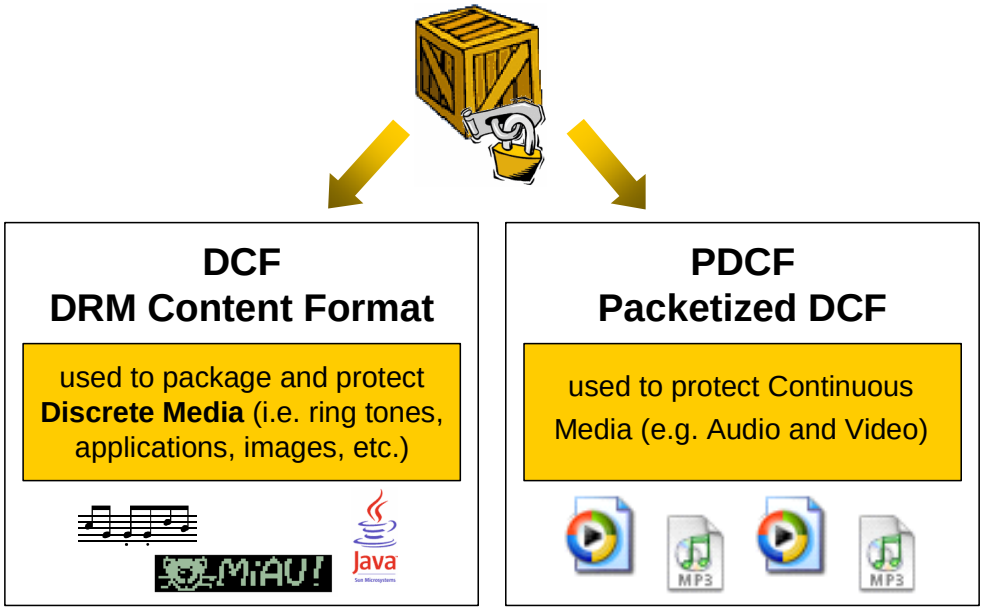


*Figure 16 DRM content format*

In addition to encrypting the Media Object the DRM Content Format supports metadata such as:

- original content type of the media object;
- unique identifier for this DRM protected Media Object to associate it with rights;
- information about the encryption details;
- information about the rights issuing service for this DRM protected media object;
- extensions and other media type dependent metadata

There are two DRM Content Format profiles:

- DCF: The first profile is used to package and protect Discrete Media (i.e. ring tones, applications, images, etc.). The Discrete Media profile allows to wrap any content in an envelope (DCF). That content is then encrypted as a single object agnostic of the contents internal structure and layout. The Discrete Media format used in OMA DRMv2 is based on the types of the ISO base media file format and on WSP types in case of OMA DRMv1. By using the ISO principles, the DCF format maintains the extensible nature of the ISO format, while keeping overhead minimal.
- PDCF: The second profile is used to protect Continuous Media (e.g. Audio and Video). Continuous media is protected in a separate profile because it is packetized and thus the profile is called the Packetized DCF (PDCF). Applications that read and parse Continuous Media are meant to work on the file on a packet-by-packet basis. To facilitate the playback of protected Continuous Media, the storage format needs to

be structured in such a way that the packets are individually protected. This structurally aware packetization is also required in order to stream Continuous Media.

# Discrete Media Profile (DCF)

The Discrete Media profile (DCF) is based on the ISO Base Media File Format data types and conventions.

The MIME type for objects conforming to the DCF format is: **application/vnd.oma.drm.dcf** and the corresponding file extension is '**.odf**'.

The ISO base media file format is structured around an object-oriented design of **boxes**. A basic box has two mandatory fields, ***length*** and ***type***. The *type* identifier is used to dynamically bind a box to a statically defined type and the *size* is an offset from start to the end of the box. The identifier is constructed from four bytes, each representing a human-readable character, thus the name *Four Character Code* (4CC).

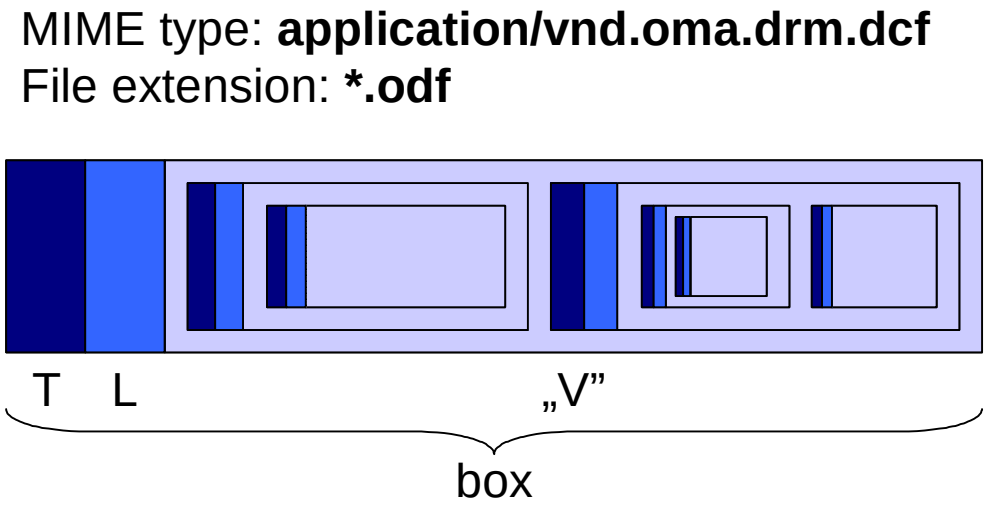


*Figure 17 ISO base media format*

The ISO base media file format allows for a file signature/brand in the file header. Files conforming to the Discrete Media profile include a brand number. The file brand is 32 bits (4 octets) wide with the hexadecimal value 0x6F646366 ('odcf'). This is followed by a four-octet version indicator, making the file brand a total of eight octets (64 bits) from the beginning of the file. The version field consists of a version major and minor numbers, two octets each. For files conforming to this OMA DRMv2 the version value is 2.0 (`0x00020000`). The Fig. 18 shows the relationship of the file brand, version and rest of the file content.

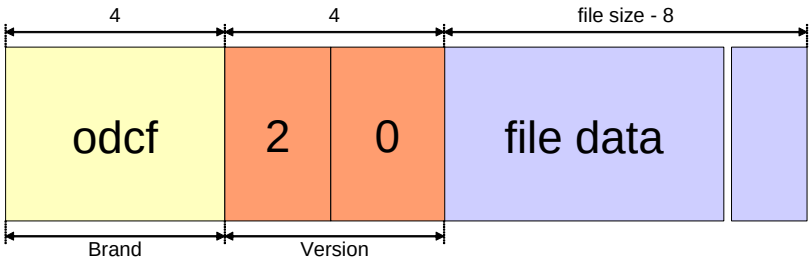


*Figure 18 DCF file header and body*

A DCF file includes at least one `OMADRMContainer` box. The `OMADRMContainer` box is a container for a single Content Object and its associated headers. It appears on the top level, i.e. it is not nested inside another data type. There may exist multiple `OMADRMContainer` boxes in a file, but one must immediately follow the file brand, and they all must be located on the top level in the nesting structure.

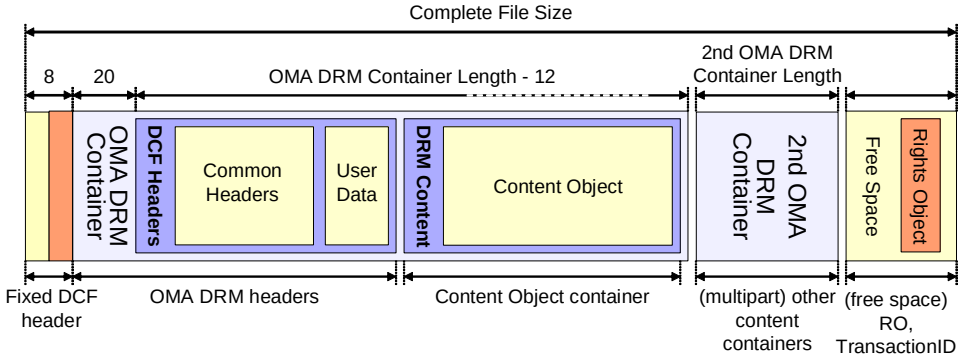


*Figure 19 DCF structure*

## DCF Headers

The DCF Headers Field contains various parameters i.e. *ContentTypeLength*, *ContentType*, *Common Headers* and *User Data*. The ***ContentType*** field indicates the original MIME media type of the Content Object i.e. what content type the result of a successful extraction of the `OMADRMContent` box represents.

### *Common Headers*

The Common Headers box defines a structure for the required headers. This box appears in both DCF and PDCF. This box includes the mandatory headers as fixed fields and provides a mechanism to insert additional headers as arbitrary name value pairs.

The most important fields present in common headers are described below.

### EncryptionMethod Field

The *EncryptionMethod* field defines how the encrypted content can be decrypted. There are three possible values for that field:

- **NULL** - No encryption for this object. NULL encrypted Content Objects may be used without acquiring a Rights Object,
- **AES_128_CBC** - AES symmetric encryption with 128 bit keys and Cipher Block Chaining (CBC) mode defined by National Institute of Standards and Technology (NIST),
- **AES_128_CTR** - AES symmetric encryption with 128 bit keys and Counter (CTR) mode defined by NIST.

Rights Issuers should take care in using NULL *EncryptionMethod* because, Null-encrypted Media Objects:

- do not have any Confidentiality protection,
- can always be used without an associated Rights Object,
- may not have any integrity protection.

### ContentID Field

The *ContentID* field contains a globally unique identifier for the Content Object. The value is a Uniform Resource Identifiers (URI) and it is the responsibility of the content author to guarantee the uniqueness of the *ContentID* within their own namespace.

If the Content Object is referenced from a DRM Rights Object, the value of the *ContentID* field MUST match the value of the referencing element of the Rights Object as defined in [DRMREL-v2]. The ContentID MUST be in the ‘cid-url’ format of [RFC2392].

### RightsIssuerURL Field

The *RightsIssuerURL* field defines the Rights Issuer URL. The Rights Issuer URL is used by the consuming Device to obtain Rights for this DRM Content.

The *RightsIssuerURL* may be empty e.g. if the Content Object is not encrypted.

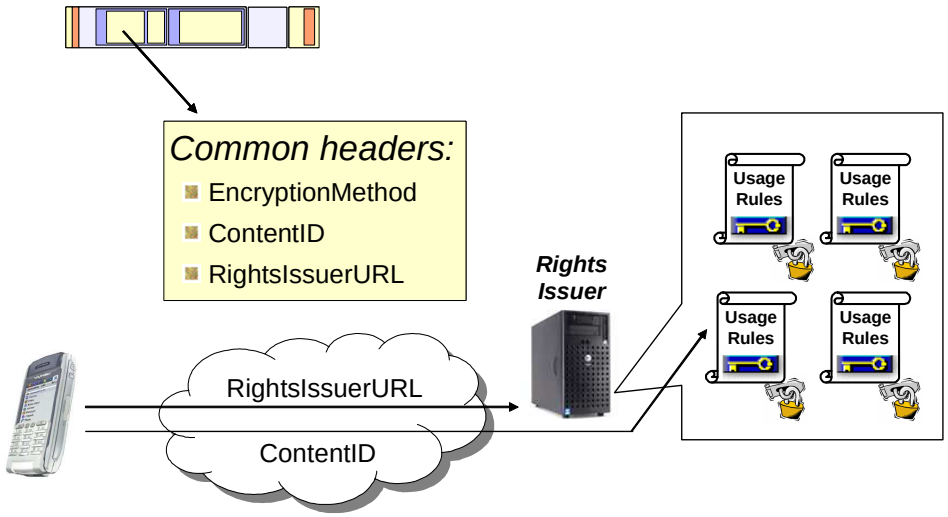


*Figure 20 Common headers box*

### *User-Data*

The user-data box is an optional container box for informative user data. The most important fields present in user-data box are described below.

#### Title

The Title box contains a descriptive name for the Content Object. The title is only informative and the device may use it e.g. to derive a filename when the DRM protected object is received and stored into a local repository.

This box can be included zero or more times using different language codes.

#### Description

The Description box contains a description of the Content Object. This text is informative and the device may display it to the user prior to acquiring Rights for the Content Object.

This box can be included zero or more times using different language codes.

#### Copyright

The Copyright box contains a copyright declaration of the organization holding the copyright of the Content Object. This text is informative and the device may display it to the user prior to acquiring Rights for the Content Object.

This box may be included zero or more times using different language codes.

#### Author

The Author box contains a textual string representing the author of the Content Object. This text is informative and the device may display it to the user prior to acquiring Rights for the Content Object.

This box may be included zero or more times using different language codes.

### IconURI

The IconURI box contains a URI where an appropriate icon for this content may be retrieved from. The device requests the object at this URI, and if an appropriate content is returned, use this as an icon associated with the content to the user.

If the DCF is a Multipart DCF, a *IconURI* may be a Content ID (CID) reference within the current file. In this case, the referenced Content Object is NULL-encrypted.

### InfoURL

The InfoURL box contains a URL where additional information can be found regarding the Content Object. The device may obtain this information prior to using the *RightsIssuerURL* field or after the Rights Object has been obtained.

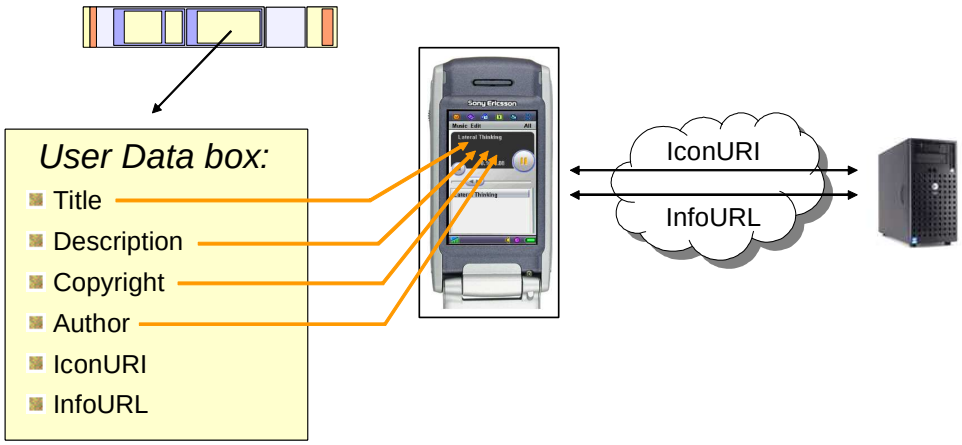


*Figure 21 User Data box*

### Content Object Box

The Content Object box includes only the data length field and data bytes for a single Content Object.

# Continuous Media Profile (PDCF)

This describes the OMA DRM key management part of the PDCF format. In the `ProtectionSchemeInfoBox`, there is space for a “black box” describing the key management governing access to the encrypted media content.

The Fig. 22 illustrates how protection information is stored in a PDCF. It is an example where only the video track is protected by placing a `ProtectionSchemeInfoBox` into the track and specifying the OMA DRM

identifier as the key management system. All tracks in a PDCF can be protected with the mechanism.

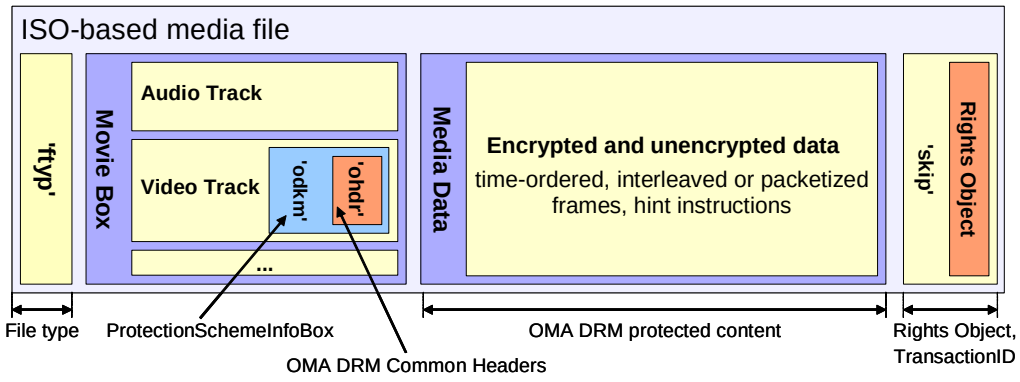


*Figure 22 Example PDCF Structure*

There is a difference between a streamable PDCF and a non-streamable PDCF. A streamable PDCF conforms to the server profile of the file format specification, and the media data is stored as packets. In a non-streamable PDCF, media data is stored as samples.

## DRM Scheme Type

The `SchemeTypeBox` includes information on which DRM system is being used to manage keys and decryption of the content. As the media file format supports also other key management systems than OMA DRM, the key management system in use is indicated by a 4CC in the *SchemeType* field.

For PDCF files conforming to OMA DRMv2, the *SchemeType* is the 4CC ‘odkm’, and *SchemeVersion* is v2.

### *Scheme Information*

The `ProtectionSchemeInfoBox` is used to carry DRM key management system specific information, thus it is only a container box. For OMA DRM, this box includes one `OMADRMKMSBox`.

There may be several instances of the `OMADRMKMSBox` in a PDCF file, and one can appear either at the movie level or exactly one per each protected track. There must not be key management boxes in both movie level and track level. Contained in the `OMADRMKMSBox` there is a single `OMADRMCommonHeaders` box. The Common headers box is exactly the same as defined for DCF.

# Rights Object

DRM defines the mechanisms how to deliver DRM Content and Rights Objects to a consuming device. Rights are used to specify the access a consuming device is granted to DRM Content. The Rights Expression Language (REL) defined by the OMA specifies the syntax and semantics of rights governing the usage of DRM Content based on the Open Digital Rights Language (ODRL).

DRM Content is consumed according to the specified rights. Therefore, the value is in the rights and not in the Content itself. Rights Objects are specified so that they only become usable on authorized devices.

Content can be stored; however, it can only be accessed if a corresponding Rights Object is available. Similarly, encrypted content can be super-distributed without unnecessarily complicating the REL; no separate distribution permissions are necessary, since DRM Content without the decryption key is of no value.

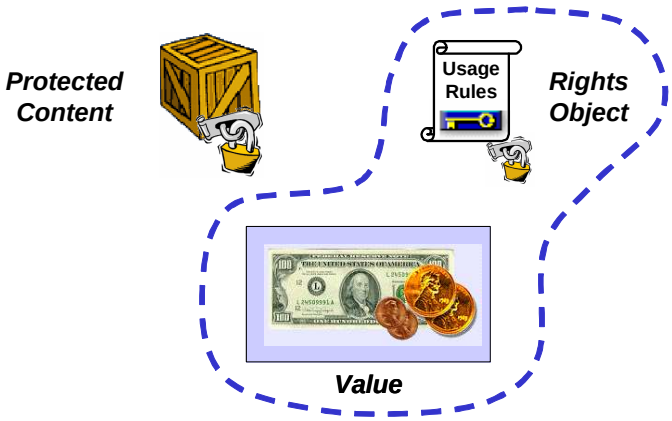


*Figure 23 Value, Content & Rights Object*

# Permissions

Usage of the DRM Content is only granted according to the permissions explicitly specified by the corresponding Rights Object. The set of permissions comprises: play, display, execute, print and export.

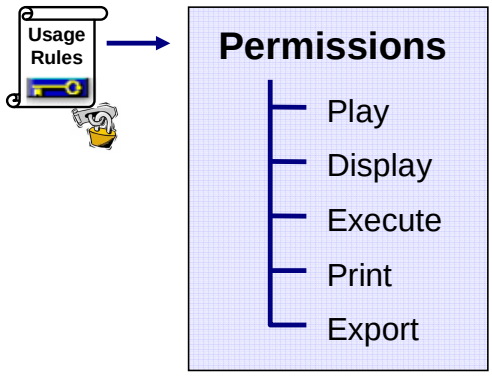


*Figure 24 Permissions*

## Play

The *play* element grants the permission to create a transient representation of audio or video Content. It contains an optional *constraint* element. If the *constraint* element is specified the DRM Agent grants play rights according to the *constraint* child element. If no *constraint* element is specified, the DRM Agent grants unlimited play rights.

An access to game content, e.g., Java games, based on the *play* permission is prohibited. In order to specify rights for Java games, the *execute* element is utilized instead.

## Display

The *display* element grants the permission to make a transient visable rendering of the Content. It contains an optional *constraint* element. If no *constraint* element is specified, the DRM Agent grants unlimited display rights.

## Execute

The *display* element grants the permission to make a transient visable rendering of the Content. It contains an optional *constraint* element. The *execute* element has the semantics of executing, i.e., invoking, DRM Content, e.g., Java games or other applications.

## Print

The *print* element grants the permission to create a fixed (i.e., static), directly perceivable representation of Content. It contains an optional *constraint* element. The *print* element has the semantics of printing, i.e., creating a hardcopy of, the DRM Content, for example, image/jpeg.

### Export

The *export* element grants export rights over DRM Content and corresponding Rights Objects. It contains a mandatory *constraint* element.

The *export* element has the semantics of exporting the DRM Content and corresponding Rights Objects to a DRM system running on another device.

A mandatory *constraint* element, which then contains a mandatory *system* element specifying to which DRM system or content protection scheme the DRM Content and Rights Objects are allowed to be exported.

## Constrains

The constraint model enhances the permission model by providing fine-grained consumption control of content.

Constraints are associated with one permission element at a time. For a permission to be granted all its constraints must be fulfilled. If a constraint is not understood or cannot be enforced by the consuming device the parent permission is invalid and must not be granted.

The *constraint* element contains the optional *count*, *datetime*, *interval*, *accumulated*, *individual*, and *system* elements.

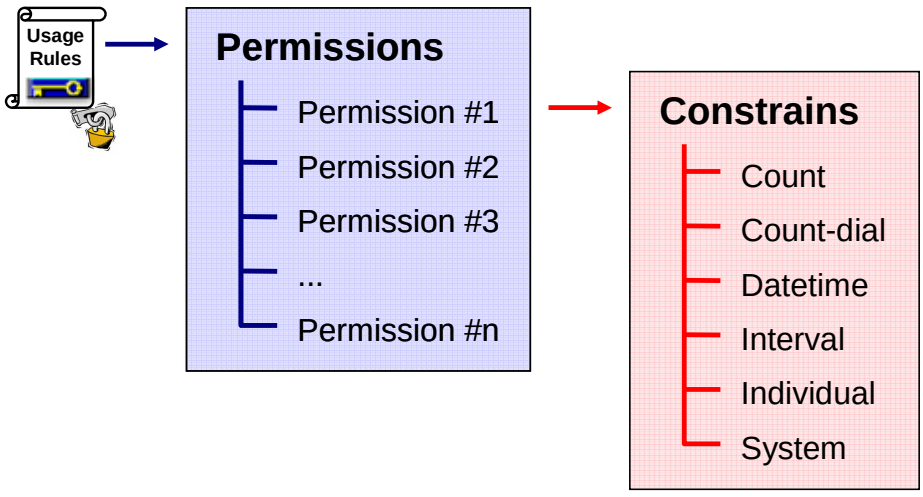


*Figure 25 Constrains*

### Count

The *count* element specifies the number of times a permission may be granted over an asset.

## Count-dial

The semantics of the *count-dial* element are as for the *count* element with the addition of an optional *dial* attribute.

The dial attribute specifies the number of seconds after which the count state specified by the value of the *count-dial* element is reduced starting from beginning to render the content.

For example, if the dial value is set to 30 and the *count-dial* constraint value is set to 5, a corresponding media object, may be rendered 5 times, while the number of remaining accesses is decremented after the content has been rendered for 30 seconds. In other words, if rendering of the content stops after less than 30 seconds, the state value of the *count-dial* element is not reduced.

## Datetime

The *datetime* element specifies the time range, respectively the time limit, for a containing permission. It contains the optional *start* and *end* elements.

If the *start* element is present, its semantics are 'not before' the specified time/date. If the *end* element is present, its semantics are 'not after' the specified time/date.

The DRM Agent of a consuming device without a time source must not grant access to DRM Content according to permissions containing the *datetime* element.

If DRM Content is rendered with the purpose of directing the user's attention to an incoming phone call or message, or to a calendar or other alarm event, the DRM Agent allows access to the DRM Content to continue until the user has taken notice of the event, for example, by answering or rejecting the phone call, or dismissing the calendar or other alarm event.

## Interval

The *interval* element specifies a period of time during which the permissions can be exercised over the DRM Content. The *interval* period begins when the associated permission is first exercised. The permission can then be exercised any number of times within the *interval* period.

The DRM Agent of a consuming device without a time source must not grant access to DRM Content according to permissions containing the *interval* element.

## Accumulated

The *accumulated* element specifies the maximum period of metered usage time during which the rights can be exercised over the DRM Content.

The DRM Agent of a consuming device without a time source must not grant access to DRM Content according to permissions containing the *accumulated* element.

### Individual

The *individual* element specifies the individual to who content is bound. It does so by binding content to the user identity specified via its *context* child element.

### System

The *system* element specifies the DRM system or content protection scheme to which DRM Content and Rights Objects can be exported, described in the mandatory *context* element. The *system* element only occures as a constraint to an *export* permission.

## Security

Security constitutes an important part of a DRM system. Even if a particular DRM system is not designed to provide the technically highest possible degree of security (because of other factors such as business models, cost of increased security vs. value of content, etc.) security must be accounted for in the REL used to express rights over DRM Content in this system. The security model enhances the agreement model. It is designed to:

- Enforce the integrity of Rights Objects,
- Ensure the controlled consumption of DRM Content,
- Enforce the integrity of the association between Rights Objcets and DRM Content.

### Rights Integrity

Integrity protection prevents illegitimately modifying the Rights Object specified for DRM Content, including adding, deleting, and modifying permissions and constraints for DRM Content, references to the DRM Content itself, and Meta information included in the Rights Object.

### Content Confidentiality

Protecting content confidentiality is an essential part of enforcing consumption control of DRM Content. Enabling an authorized party to consume content is similar to granting this party access to the confidential content. In other words, a party authorized to consume content is let into the

exclusive circle of parties deemed trustworthy enough to access the protected content. This concept is realized by:

- encrypting the DRM Content,
- sharing the key required to decrypt the DRM Content only with those parties that are authorized to consume the content.

Content is encrypted using a symmetric algorithm (AES), i.e., the key used for decryption can be derived from the key used for encryption. Thus, henceforth, the key will be referred to as *Content Encryption Key*, or short *CEK*. Encrypting the content defers content confidentiality to controlling the confidentiality of the CEK. Now, the security of the DRM system relies on the control of the CEK that must be kept secret from all unauthorized parties.

The CEK is not encrypted and thus its confidentiality is dependent on the delivery mechanism.

The security elements present in the RO are used to achieve the desired level of security.

### *KeyInfo*

The *KeyInfo* element is the starting point for all consumption control, i.e., content encryption, functionality. It contains the *KeyValue* element.

The *KeyInfo* element associates the corresponding protection with the asset governed by the rights.

### *KeyValue*

The *KeyValue* element contains the content encryption key required for content consumption in plain.

## Rights Object - DRM Content Association Integrity

The ability to replace the DRM Content governed by rights amounts to the ability of changing the rights itself. Thus, the integrity of the association between a Rights Object and the corresponding DRM Content must be protected as much as the specified rights.

A reference to a piece of DRM Content is established via the *uid* element. Enforcing the integrity of the association between Rights Object and DRM Content is handled similarly to enforcing the integrity of Rights Objects i.e., by signing Rights Objects. This prevents tampering with the content identifier in the Rights Objects. It does not, however, prevent from modifying the corresponding identifier in the DRM Content.

It is possible to provide a way of securing the content-end of the Rights Object - DRM Content association without having to sign the DRM Content. Instead, a hash of the (encrypted) DRM Content is included in the Rights

Object. Since this hash value is part of the signed Rights Object, it is as safe from being tampered with, as is the *uid* element in the Rights Object referencing the DRM Content. The integrity of the content-end is guaranteed by the very characteristics of the hash itself: any modifications to the DRM Content automatically invalidate the hash value inside the Rights Object. Please note that including the hash of the content in the Rights Object only ensures the integrity of the Rights Object - DRM Content association if the integrity of the Rights Object is protected also, e.g., by signing it.

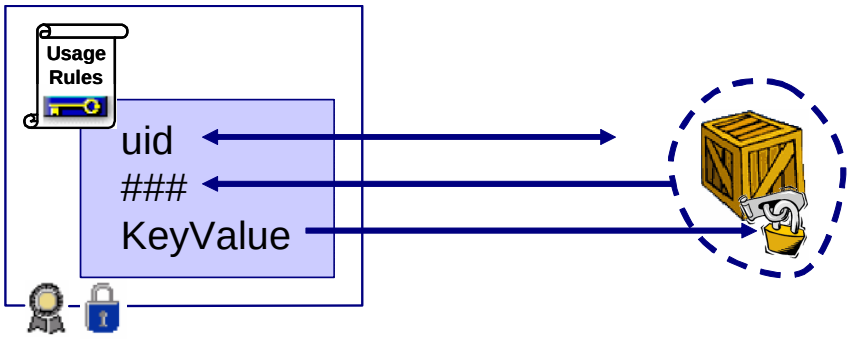


*Figure 26 Security elements*

# Use Cases

OMA DRM is designed to be flexible and support a wide variety of different business and usage models. This section outlines some use cases covered by the specifications. It is not an exhaustive list.

## Basic Download

### Basic Pull Model

One model for content distribution is using OMA OTA Download mechanisms. The client would launch a browser and connect to a Content Issuer portal. The user would evaluate the content offerings from this portal and make a decision on specific items of content to be downloaded. Once the DRM Content is downloaded, the client can connect to the Rights Issuer portal and engage in the Rights Object Acquisition Protocol to acquire the associated rights.

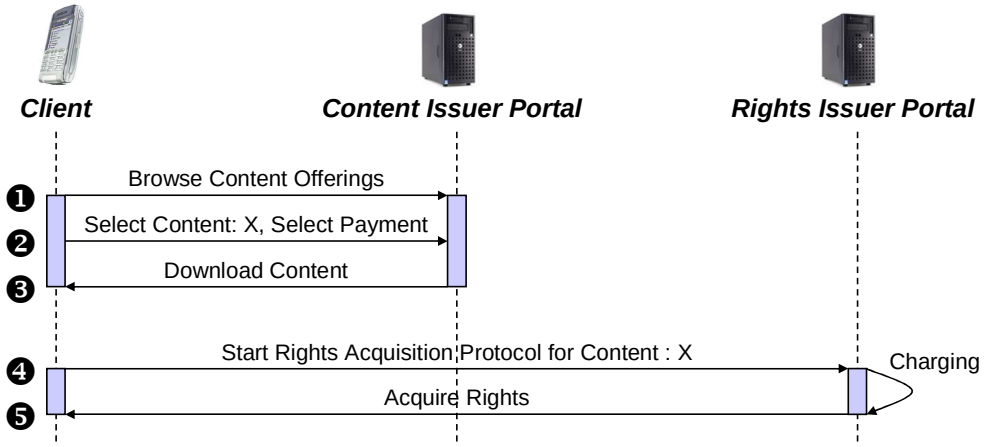


*Figure 27 Basic pull model*

❶ The client initiates a browsing session with the Content Portal.

❷ The client selects the specific content from the content offerings on the portal. In addition, the client may select the payment mode during this session.

❸ The client downloads the DRM Content from the portal to local storage.

❹ The client looks up the Rights Issuer URL within the DRM Content headers and initiates a connection to the Rights Issuer portal. And engages in the Rights Object Acquisition Protocol.

❺ The client, at the successful completion of this protocol, acquires the Rights Object associated with the DRM Content.

## Content Push

The content issuer/rights issuer may have some previous knowledge of a user and a particular DRM Agent, so that content and a Rights Object can be correctly formatted and packaged for delivery. For example, the user may have registered to receive a daily background image to her phone, or the hit song of the week. In this case the process would go through the same steps as above, but delivery of DRM Content and Rights Objects would be over WAP Push or MMS.

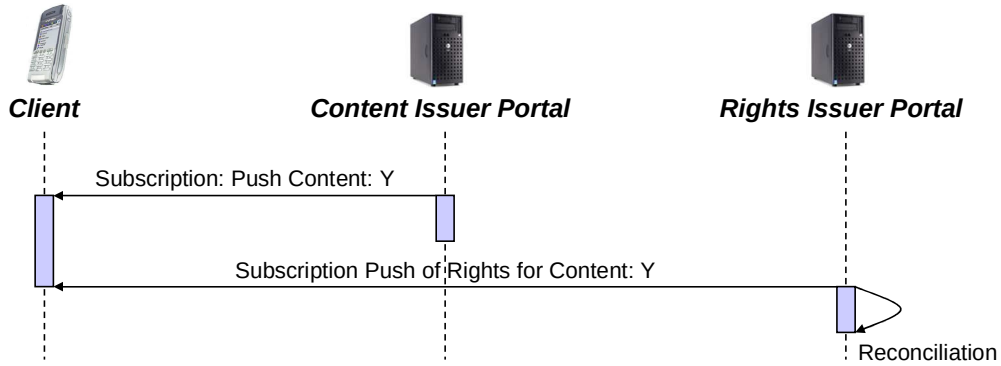


*Figure 28 Content push model (scenario 1)*

Scenario shown in this picture is the subscription push of both content and rights. In this model, the client has an established subscription and charging agreement with the Rights Issuer in place. As a result of this, the Rights Issuer can push both DRM Content and Rights Objects to the clients on a regular interval.

Scenario shown in Fig 29 is the subscription based push that in turn initiates a pull of the Rights Object from the Rights Issuer Portal. The DRM Content is delivered with the 'silent' header ("in-advance") and the Client, on reception of the content, connects to the Rights Issuer to trigger the Rights Object Acquisition Protocol. On completion of this protocol successfully, the Rights Object is issued to the client.

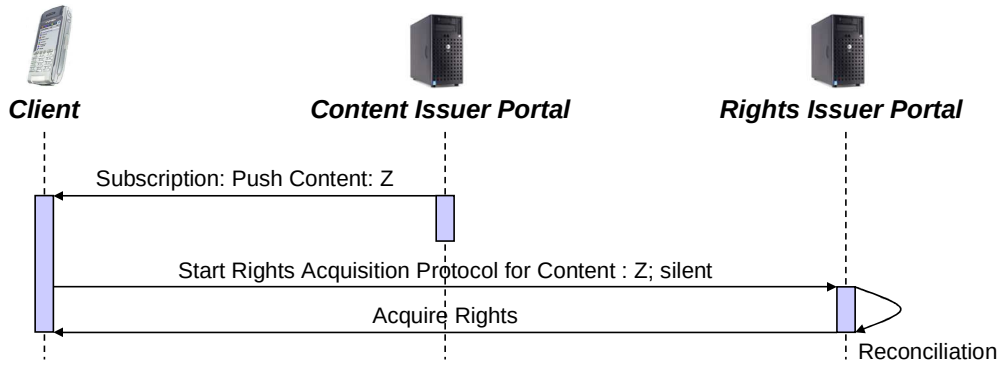


*Figure 29 Content push model (scenario 2)*

## Push-initiated Pull

In this case, the content issuer/rights issuer has no previous knowledge of a user or the target DRM Agent, but still wishes to send content. For example, one user may buy some content as a gift to another user. In this case, the content provider does not yet know what content is suitable for the receiving device, how trusted the receiving DRM Agent is, and so on. Instead of pushing DRM Content directly, a link to the content can be sent. Following

the link will take the receiving user to a specific location, and then the procedure continues as in the basic pull model.

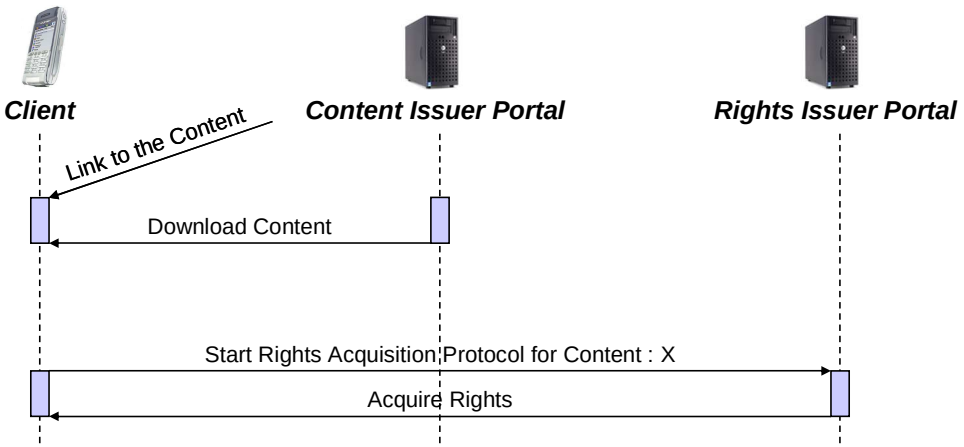


*Figure 30 Push-initiated pull*

# Super Distribution

A given client who has downloaded content from a Content Issuer can in turn distribute this DRM Content to other devices using various networked links as well as removable media. This DRM Content is encrypted and is not usable by the receiving device/user until the associated rights are acquired for the content. The device that receives this super-distributed content will discover the Rights Issuer URL within the DRM Content headers and use this information to connect to the Rights Issuer portal to acquire the rights. The interaction diagram below illustrates this model of content distribution and the related flow of events amongst the significant actors.

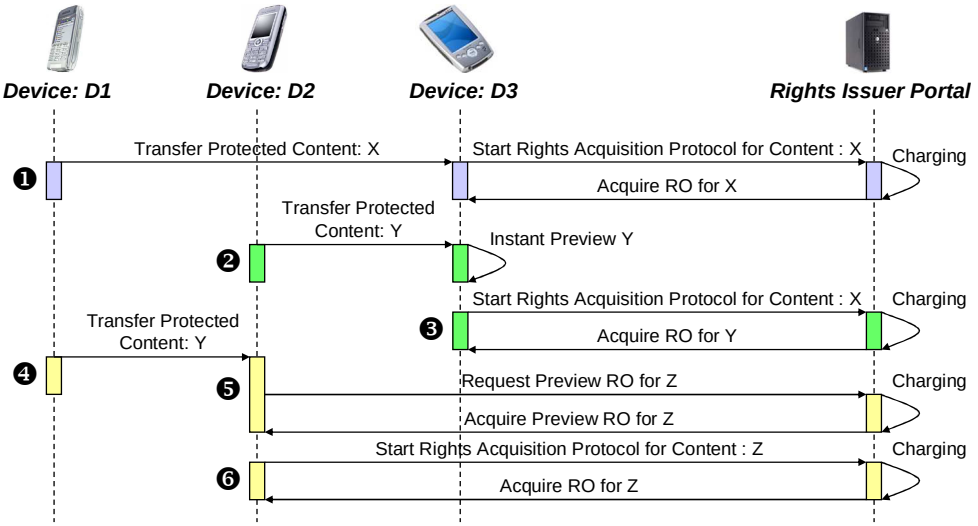


*Figure 31 Super distribution*

❶ Device D1 has previously received some DRM Content and has it stored locally. Device D1 wants to share this DRM Content with Device D3, and as a result, transfers this to D3 using local connectivity or removable media. Device D3, on reception of this DRM Content, discovers the Rights Issuer URL from the DRM Content headers and initiates a Rights Object Acquisition Protocol session with the Rights issuer. On completion of this protocol and appropriate payment arrangements, the device D3 obtains the Rights Object associated with DRM Content X. Now, the user of device D3 is able to use this content.

❷ Device D2 transfers DRM Content Y to device D3. This DRM Content Y has the 'preview' headers and is able to provide an 'instant preview' for the content within it. The device D3 can make the 'preview' available to the user and the user can make a decision regarding the content purchase.

❸ Once the user of device D3 has decided to purchase the rights for content Y, it initiates the Rights Object Acquisition Protocol with the Rights Issuer. On successful completion of this protocol, the device D3 obtains the Rights Object for DRM Content Y.

❹ Device D1 transfers DRM Content Z to device D2.

❺ On reception of this DRM Content Z, the device D2 discovers that this content can provide a preview if the device obtains a preview Rights Object. As a result, the device D2 connects to the Rights Issuer and obtains the Rights Object to enable a preview. Rights Objects provided are full-fledged Rights Objects, the only difference being that the permissions and constraints are specified to just enable a preview. This may or may not result in charging, depending on the business model.

❻ Once the user decides to purchase the rights, the device D2 starts a Rights Object Acquisition Protocol session to acquire rights for content Z. On successful completion of the protocol, the Rights Object for Z is obtained by the device.

# Streaming Media

The two previous examples assume that content is packaged and delivered in its entirety. Alternatively, content may be packetised and delivered as a stream.

For distributing protected streams, the streaming token[1] is acquired from the Content Issuer portal and the access to the streams is governed by the associated Rights Object. The client, after receiving the session headers, can connect to the Rights Issuer and acquire the necessary Rights Object, which in turn will provide the necessary information for the client to be able to

[1] A streaming token is a piece of data that the streaming player uses to determine the location of streaming media, possibly also to determine properties of the streaming session or streams, and to set up and start the delivery of streaming media. For the 3GPP Packet-Switched Streaming Service for example, this can either be a SMIL presentation, an SDP session description, or an RTSP URL.

decode the streams and render the content. The interaction diagram below illustrates the flow of events and the technical elements necessary for this solution.

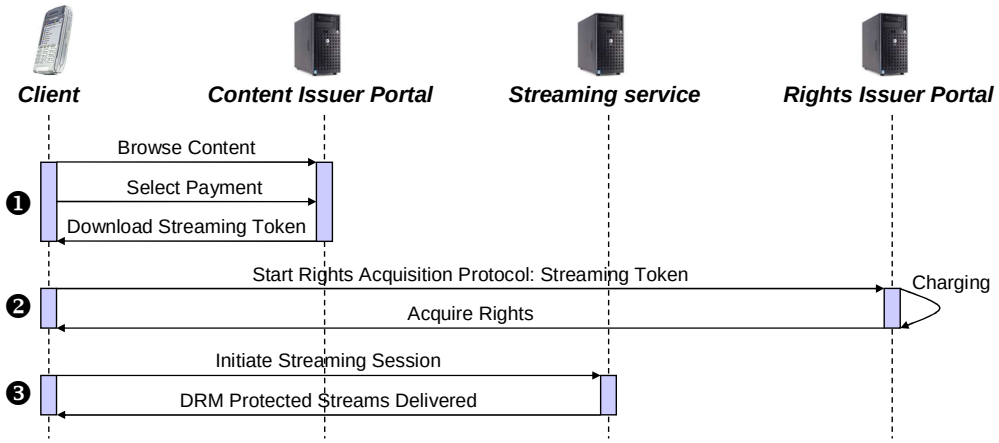


*Figure 32 Streaming media (scenario 1)*

❶ The client connects to the Content Issuer portal and browses for content of interest. Client selects the streaming service of interest, possibly indicates the payment mode, and downloads the streaming token.

❷ The Client requests rights by connecting to the Rights Issuer and initiating the Rights Object Acquisition Protocol to acquire the rights for the streamed content. On successful completion of the protocol, the client obtains the Rights Object for the streaming service.

❸ The Client connects to the Streaming server and initiates the streaming session. After the stream is initiated, the Client will have the stream properties available. The DRM properties will be included in these stream properties (except for the case of an SDP description token, where the properties are already contained in the token).

### *Push streaming token*

Scenario, where the streaming token is pushed to the DRM Agent is possible as well, see Fig. 33.

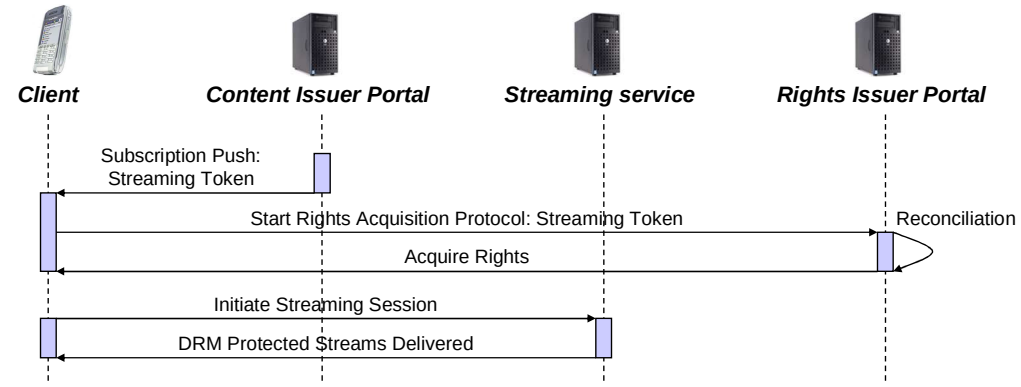

*Fig. 33 Streaming media (scenario 2)*

### *Push rights*

Another mode of delivering streaming service is when the rights are delivered in advance or along side the streaming token. The client can then connect to the streaming server and initiate the streaming session. The DRM Agent will have rights so the client will be able to immediately start the streaming session instead of going through step 2 above.

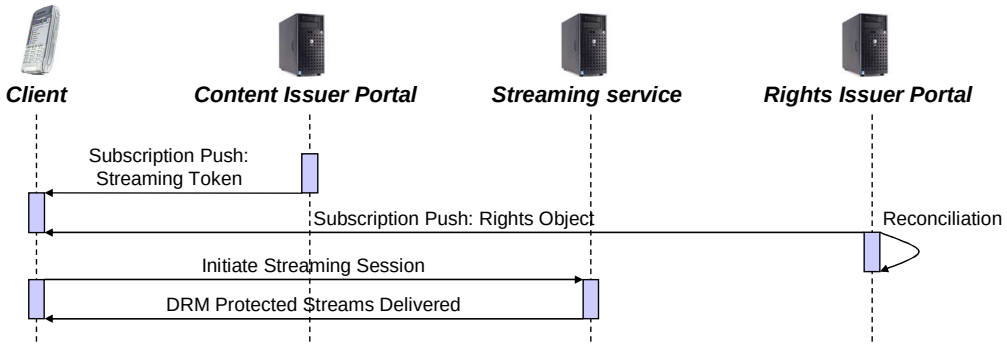


*Figure 34 Streaming media (scenario 3)*

# Domains

The basic model of OMA DRM involves binding Rights Objects and content encryption keys to a specific DRM Agent. Domains expand this notion, allowing a rights issuer to bind rights and content encryption keys to a group of DRM Agents instead of just a single DRM Agent. Users may then share DRM Content off-line between all DRM Agents belonging to the same domain.

Using this feature a rights issuer may provide new services such as enabling users to access DRM Content from several devices that they own. Other new scenarios enabled by the Domain concept include support for Unconnected Devices where users purchase DRM Content and rights via one device (e.g. a PC) for later use on another device (e.g. a portable player with no wide area network connectivity).

It is entirely up to the rights issuer if they wish to provide services based on domains, and it is entirely under rights issuer control what DRM Agents form part of a particular domain.

Domain is created, managed and administered by a Rights Issuer. Once the domain is formed and the devices are enrolled in the domain, content and rights distributed to any of the devices in the domain can be shared among the other devices in the domain without connecting back to the Rights Issuer. Alternatively, a device can join a desired domain on reception of content that is targeted for a domain.

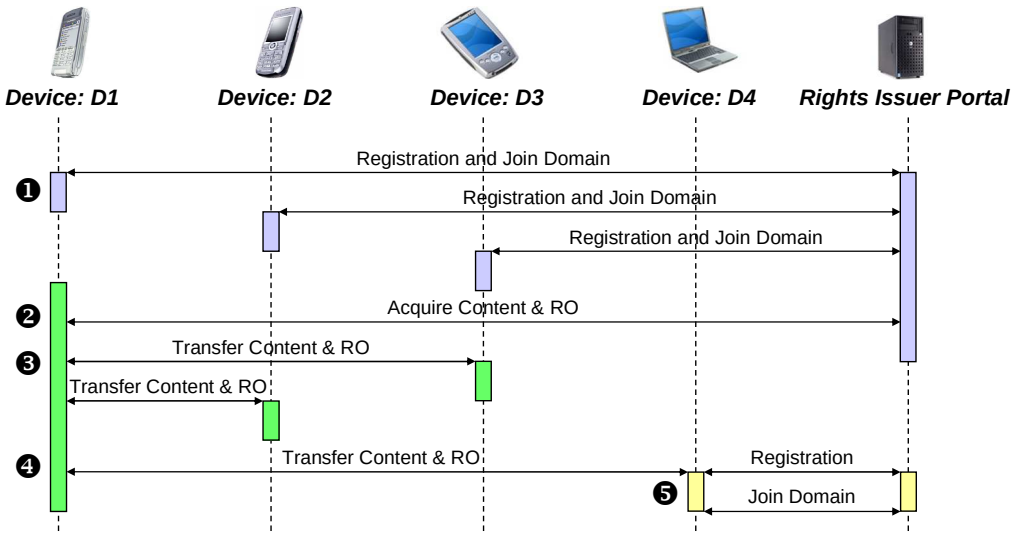


*Figure 35 Domains*

❶ In the scenario illustrated above, each of the devices D1, D2, and D3 connect to the RI and complete the registration and join a domain DM1.

❷ At a later time, device D1 connects to the RI and acquires content DCF1 and the associated domain RO for the DCF, DRO1. Now since the device D1 is part of the domain DM1, the content and rights are usable on this device.

❸ Subsequently, the device D1 forwards the content and the associated domain RO to the other devices D2, & D3.

❹ Since D2 & D3 are part of the domain DM1, the content and associated rights are immediately usable on those devices without connecting to the RI.

❺ At a later time, content is also forwarded to device D4. This device D4 has not joined the domain DM1. As a result, the content is not usable on this device. The user can choose to connect to the RI and join the domain DM1 to gain access to this content. Since the domain management is conducted by the RI, the RI can explicitly decide on the composition of the domain and decide on whether D4 can join the domain or not.

# Export

DRM Content may be exported to some other DRM system, for use on devices that are not OMA DRM compliant but support some other DRM mechanism – e.g. export to copy protected media. The rights issuer may limit export only to specific external DRM systems.

The capabilities of the other DRM system can be provided to the Content Portal so the downloaded content and rights are compatible with the target DRM system. This downloaded content is stored and managed on the original device for later export to a consuming device. OMA DRM does not define how to translate from OMA DRM to other protection mechanisms. It

merely allows Rights Issuers to, if they wish, express permission for DRM Agents with such a capability to do so.

The interaction diagram below illustrates the flow of content and rights in this model.

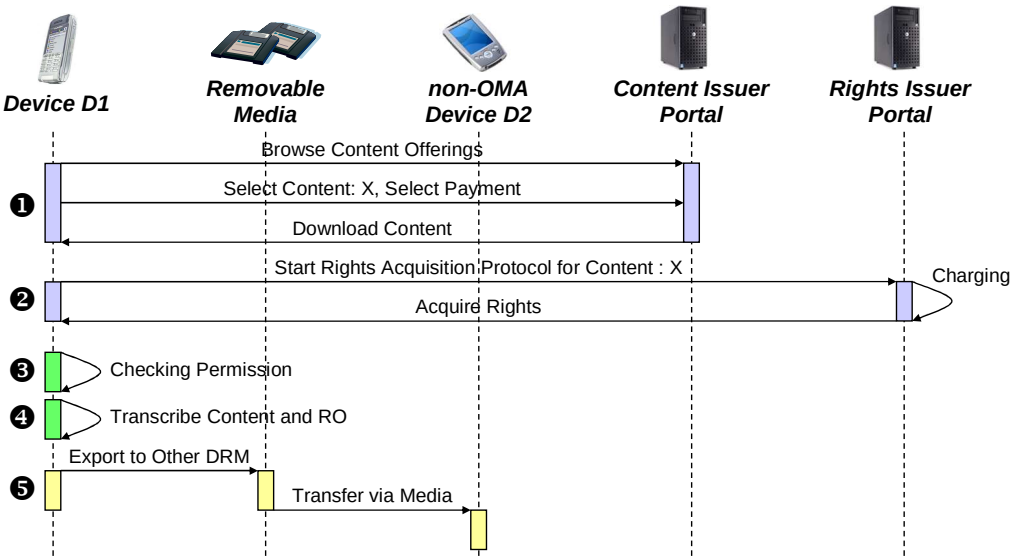


*Figure 36 Export*

❶ The client initiates a browsing session with the Content Portal. The client selects the specific content for future export from the content offerings on the portal. The content should be suitable for the target DRM system. Subsequently, the client downloads the DRM Content from the portal to local storage.

❷ Device D1 now connects to the RI to acquire the rights for the content. The rights issued are compatible with the usage rules of the target DRM system.

❸ The User wants to transfer the DRM Content to Device D2 that has a different (non-OMA) DRM system using local connectivity or removable media. The OMA DRM Agent checks the permissions described in the Rights Object to determine whether the Rights Issuer allows the content to be exported to the target DRM system, whether its content type is appropriate, and whether its usage rules are compatible with the target DRM system.

❹ The OMA DRM Agent transfers the decrypted content and Rights Object to the other (non-OMA) DRM Agent. The other DRM Agent transcribes the compatible rights to the other DRM usage rules according to the specific rules defined by the Rights Issuer and the other DRM system to maintain consistency with the original Rights Object.

❺ The User is now able to securely use this content on Device D2.

# Unconnected Device Support

OMA DRM enables a Connected Device to act as an intermediary to assist an Unconnected Device to purchase and download content and Rights Objects. This functionality enables, for example, a portable, mobile device that does not have inherent network connectivity to acquire DRM Content and associated Rights Objects. This functionality builds on the Domain concept as described earlier.

For example, a user has an OMA DRM compliant portable device (Unconnected Device) that has no wide area network connectivity, and an OMA DRM compliant mobile device (Connected Device) that has wide area network connectivity. She uses the Connected Device to browse and purchase DRM Content, and download the DRM Content to the Connected Device.

If the user wishes to render the DRM Content on the Unconnected Device then the DRM Agent on the Connected Device requests and downloads a Domain Rights Object from the rights issuer. The DRM Agent on the Connected Device then embeds the Domain Rights Object in the DCF. The DCF (with embedded Domain RO) can then be transferred to the Unconnected Device using an appropriate protocol over a local connectivity technology e.g. OBEX over IrDA, Bluetooth or USB.

Using intermediaries in this way can be useful if the Unconnected Device has a limited UI. Both the Connected and Unconnected Device must be OMA DRM compliant. Since the Unconnected Device support is built upon the Domain concept then the Unconnected Device must also belong to the same Domain as the Connected Device. In order to join the Domain the Connected Device can provide network connectivity to enable the Unconnected Device to perform the steps required to join a Domain.

The interaction diagram below illustrates the flow of content and rights in this model.

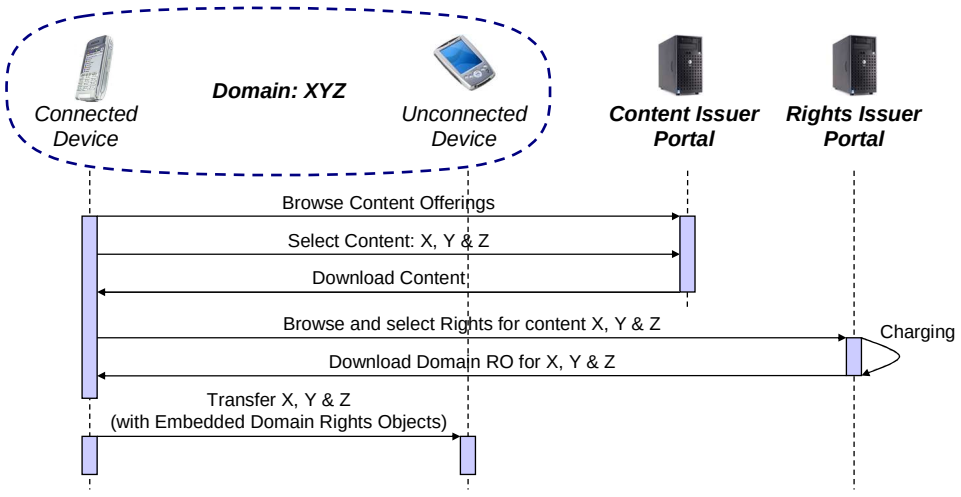


*Figure 37 Unconnected device support*

❶ The Connected Device connects to the Content Issuer portal. After a browsing session to select the content, The DRM Content X, Y, & Z are downloaded to the Connected Device.

❷ The Connected Device now connects to the RI to acquire Domain Rights Objects for the content X, Y, & Z. The Connected Device embeds the Domain Rights Objects inside the corresponding DCF.

❸ At a later time, the Connected Device transfers the DRM Content X, Y, & Z (with embedded Domain Rights Objects) to Unconnected Device over a local connection.

## Backup

DRM Content can be stored safely on removable media, in a network store, or in some other form of storage. DRM Content is stored in encrypted form, and so can only be accessed by a particular target DRM Agent using an associated Rights Object.

Rights Objects can be stored for backup purposes if the Rights Object only contains stateless permissions. The security model ensures that the Rights Object is protected and can only be accessed by the intended DRM Agent – even if a Rights Object is stored off-device, it will still only allow the intended DRM Agent to access associated DRM Content.

Some permissions require maintenance of state by the DRM Agent, for example a limited number of plays. Such Rights Objects cannot be stored off-device, as this might result in loss of state information - e.g. current number of plays. A lost or damaged Rights Object may still be restored via the rights issuer by requesting a new Rights Object.

# Trust and Security Model

The fundamental challenge facing any DRM solution is how to ensure that permissions and constraints associated with DRM Content are enforced. The main threat comes from unauthorised access to DRM Content beyond what is stipulated by the associated Rights Objects, or creation of illegal copies and redistribution of valuable content such as music and games.

Rights Objects and DRM protection are enforced at the point of consumption. This is modelled in the OMA DRM specifications by the introduction of a DRM Agent. The DRM Agent embodies a trusted environment within which DRM Content can be securely consumed. Its role is to enforce permissions and constraints and to control access to DRM Content.

# Overview

The basic steps for distributing DRM Content can be summarised as follows:

1. Content packaging: Content is packaged in a secure content container (DCF). DRM Content is encrypted with a symmetric content encryption key (CEK). Content can be pre-packaged, i.e. content packaging does not have to happen on the fly.

   Although not required by the OMA DRM specifications or the OMA DRM architecture, it is recommended that the same CEK is not used for all instances of a piece of content. Using the same CEK for all content instances would pose a greater risk if a single device was to be hacked and a CEK stored on that device exposed. Using a different CEK for different deliveries or different devices will limit this risk.

2. DRM Agent authentication: All DRM Agents have a unique private/public key pair and a certificate. The certificate includes additional information, such as maker, device type, software version, serial numbers, etc. This allows the content and rights issuers to securely authenticate a DRM Agent. Any privacy aspects with releasing such information are addressed in the technical specifications.

3. Rights Object generation: A Rights Object is an XML document, expressing the permissions and constraints associated with the content. The Rights Object also contains the CEK – this ensures that DRM Content cannot be used without an associated Rights Object.

4. Rights Object protection: Before delivering the Rights Object, sensitive parts are encrypted (e.g. the CEK), and the Rights Object is then cryptographically bound to the target DRM Agent. This ensures that only the target DRM Agent can access the Rights Object and thus the DRM Content. In addition, the RI digitally signs the RO.

5. Delivery: The RO and DCF can now be delivered to the target DRM Agent. Since both are inherently secure, they can be delivered using any transport mechanism (e.g. HTTP/WSP, WAP Push, MMS). They can be delivered together, e.g. in a MIME multipart response, or they can be delivered separately.

# Trust Model

The DRM Agent has to be trusted by the rights issuer, both in terms of correct behaviour and in terms of a secure implementation. In OMA DRM, each DRM Agent is provisioned with a unique key pair, and an associated certificate, identifying the DRM Agent and certifying the binding between the agent and

this key pair. This allows rights issuers to securely authenticate the DRM Agent using standard PKI procedures.

The information in the certificate enables the Rights Issuer to apply a policy based on its business rules, the value of its content, etc. For example, a rights issuer may trust certain manufacturers, or it may keep an updated list of DRM Agents that are known to be good or bad according to some criteria defined by the rights issuer. It is also possible for a group of stakeholders to establish a joint authority identifying trusted DRM Agents, with legally binding compliance rules.

Revocation in this model amounts to not distributing content any more to DRM Agents that are no longer considered trusted. What constitutes a trusted DRM Agent depends on the policy and business model of rights issuers. For example, if a hack or a fault compromises a whole class of devices, a rights issuer may decide to stop distributing new content to all devices of that type or class. This is a worst-case scenario. At the other end of the spectrum, maybe there is a known bug in devices of a certain type, but the risk of content leaking is relatively small. In such cases, content and rights issuers may choose to continue to deliver content to existing devices, and instead let manufacturers correct the problems in future versions. Either way, the secure mechanism for authenticating DRM Agents enables rights issuers to enforce such policies.

# Content Protection

The DRM Content Format (DCF) is a secure content package for encrypted content, with its own MIME content type. In addition to the encrypted content it contains additional information, such as content description (original content type, vendor, version, etc.), rights issuer URI (a location where a Rights Object may be obtained), and so on. This additional information is not encrypted and may be presented to the user before a Rights Object is retrieved.

Since a DCF is inherently secure, it can be transported using any transport protocol, e.g. in an HTTP response or in an MMS message. It can be stored for back-up on any kind of storage, e.g. removable media or a networked PC. It can be copied and sent to another DRM Agent, where a Rights Object may be acquired for use on the receiving device (superdistribution).

The content encryption key needed to unlock DRM Content inside a DCF is contained within a Rights Object. Thus it is not possible to access DRM Content without a Rights Object. DRM Content can only be used as specified in a Rights Object.

OMA DRM includes a mechanism allowing a DRM Agent to verify the integrity of a DCF, protecting against modification of the content by some unauthorised entity.

# Rights Object

Rights Objects are used to specify consumption rules for DRM Content. The Rights Expression Language (REL) defined by OMA DRM specifies the syntax (XML) and semantics of permissions and constraints governing the usage of DRM Content. An instance of a rights document is called a Rights Object, and has its own MIME content type.

Rights Objects are made up of permissions (e.g. play, display and execute) and constraints (e.g. play for a month, display ten times) – see [OMA REL]. Rights Objects may also include constraints that require a certain user (user identity) to be present when the content is used. These permissions and constraints, along with other information embodied in the Rights Object, (e.g. copyright information) may be presented to the user. The Rights Object also governs access to DRM Content by including the content encryption key (CEK).

A single Rights Object may be associated with multiple pieces of DRM Content. Further, it is possible to assign different permissions to different components of a composite object.

Conversely, a single piece of DRM Content may be associated with multiple Rights Objects. If there are multiple Rights Objects associated with a piece of DRM Content, each Rights Object is treated individually – Rights Objects are not combined. This means that at any one time, there may be more than one Rights Object whose constraints are satisfied. When this is the case, the DRM Agent selects one to enforce. This selection may be made automatically by the DRM Agent based on some selection criteria, e.g. picking the least restrictive Rights Object, or it may be done based on user interaction.

# Rights Object Protection

A Rights Object is protected using a rights encryption key (REK). The REK is used to encrypt sensitive parts of the Rights Object, such as the CEK. In addition, the RO is digitally signed by the RI.

During delivery, the REK is cryptographically bound to the target DRM Agent. In this way only the target DRM Agent can access the Rights Object, and thus the CEK.

Since a protected Rights Object is inherently secure, it can be copied and stored off-device for backup purposes. Some permissions require maintenance of state by the DRM Agent, for example a limited number of plays. Rights Objects containing such permissions cannot be copied or stored off-device, if this would result in loss of state information - e.g. current number of plays.

## Other Security Aspects

The building blocks described above address the main security issues of protecting content and Rights Objects from unauthorised access. In addition, OMA DRM addresses a number of other security aspects, including:

- Rights Issuer Authentication

  Rights issuers are required to authenticate themselves to the DRM Agent during delivery of Rights Objects. This gives some level of assurance about the authenticity of the rights issuer.

- Rights Object Replay Protection

  An example of Rights Object replay would be if an intermediary intercepts a Rights Object with a limited number of plays during delivery to the DRM Agent. When the rights run out on the DRM Agent, the intercepted Rights Object might be delivered again (replayed) from the intermediary. OMA DRM prevents this and similar attacks from occurring.

- DRM Time

  Some constraints (absolute time constraints), as well as some aspects of the delivery protocol for Rights Objects, rely on the DRM Agent having a secure time source. DRM Time in the context of the OMA DRM specifications means accurate as well as not changeable by users. Since users are not able to change the DRM AgentTime, the OMA DRM specifications provide mechanisms for the DRM Time to be synchronised when necessary, e.g. if DRM Time is lost after prolonged power failure. Due to the limited capabilities of some Unconnected Devices, Unconnected Device may not support a real time clock and therefore will not support DRM Time. Within OMA DRM Connected Devices must support DRM Time.

# Use Cases

The DRM Trust Model required by this specification is based on the Public Key Infrastructure (PKI). In this model, typically, there are groups of principals, verifiers and one or more authentication authorities recognized and trusted by both. A single entity can play both as a principal and a verifier depending on the needs of the solution being crafted. The overall purpose of the infrastructure is to enable a verifier to authenticate the identity and other attributes of a principal when they communicate over an open, unsecured network. In such a system, typically, the verifier does not have to maintain any sensitive information about the principals it interacts with, for the

purposes of authentication. In addition, the CA is not directly involved in transactions between principal and the verifier.

The primary entities of the trust model as it is specified in this specification are the CAs, Devices and Rights Issuers. The authentication and key transfer protocols developed require Rights Issuer to be able to authenticate the Device and the Device to be able to authenticate the Rights Issuer. Mutual authentication is accomplished by the Rights Object Acquisition Protocol (ROAP).

- It is assumed that devices are provisioned (either at manufacturing time or later) with Device public and private keys and associated certificates signed by an appropriate CA. A Device manufacturer could be a CA by itself in order to sign the certificates.
- The Device can be provisioned with more than one certificate. Based on the certificate preferences expressed by the Rights Issuer, the Device has to provide an appropriate certificate.
- It is also required that the Device stores the private keys in local storage with integrity and confidentiality protection.
- The Rights Issuers are also provided with public and private keys and certificates. The certificates would be signed by a CA. The certificate chain is presented to the Device at the time of the authentication protocol so that the Device can validate the certificate path.
- There could be multiple CAs in this system. This specification does not mandate a specific trust model such as a hierarchical trust model or a bridge trust model. The exact nature of these trust models is left up to the marketplace decisions.
- The ROAP protocol also requires that the CA who signs the Rights Issuer certificates runs an OCSP responder for use during the execution of the protocol.
- The CAs are also required to define the appropriate certificate policies to govern the use of the issued certificates.

Irrespective of the deployment configurations, the Media Objects are packaged and delivered to users in a protected and controlled manner. The content issuer delivers DRM Content from a portal to the Device. The Rights Issuer authenticates the Device and provides the necessary Rights Objects so that the content can be used. The DRM Agent on the Device participates in the authentication protocol and implements the necessary security and trust elements so that the Rights Objects are utilized in a conforming manner.

The Rights Objects govern the usage of the DRM Content by specifying the permissions and constraints as needed. These Rights Objects are also protected by encryption such that only the target devices obtain access to the DRM Content.

Within the OMA DRM, the DRM Content and Rights Objects are separate entities. But, they are logically associated with each other and this

association is protected. The DRM Content and Rights Objects can arrive at the Device in a number of ways – over the air, through local connectivity, through both push and pull mechanisms, etc. The system does not specify any ordering or sequence for the delivery of these objects to the Device either.

One of the fundamental functions of the DRM Agent is to enforce the permissions specified in the Rights Object during content usage. It is required that the secrets and keys that are part of the system security are protected and handled such that un-authorized use is avoided.

The OMA DRM specifies the content formats, rights expression language, authentication/authorization protocols, and protection mechanisms. OMA DRM also specifies how DRM Content and Rights Objects can be transported to devices using a number of transport mechanisms. The following sections describe some example models for content distribution and consumption that are supported by these specifications.

# Acronyms and Abbreviations

| | |
|---|---|
| **3G** | 3-rd Generation |
| **3GPP** | 3rd Generation Partnership Project |
| **AES** | Advanced Encryption Standard |
| **CBC** | Cipher Block Chaining |
| **CEK** | Content Encryption Key |
| **CTR** | Counter |
| **DCF** | DRM Content Format |
| **DMP** | Discrete Media Profile |
| **DRM** | Digital Rights Management |
| **EMS** | Element Management System |
| **GPRS** | General Packet Radio Service |
| **HTTP** | Hypertext Transfer Protocol |
| **IP** | Internet Protocol |
| **IRDA** | Infrared Data Association |
| **ISO** | International Organization for Standardization |
| **MIME** | Multipurpose Internet Mail Extensions |
| **MMS** | Multimedia Messaging Service |
| **OBEX** | Object Exchange |
| **ODRL** | Open Digital Rights Language |
| **OMA** | Open Mobile Alliance |
| **PDA** | Personal Digital Assistant |
| **PDCF** | Packetized DRM Content Format |
| **PKI** | Public Key Infrastructur |
| **PSS** | Packet Streaming Service |
| **REL** | Rights Expression Language |
| **ROAP** | Rights Object Acquisition Protocol |
| **RTSP** | Real Time Streaming Protocol |
| **SDP** | Session Description Protocol |
| **SMS** | Short Message Service |
| **URI** | Uniform Resource Identifier |
| **USB** | Universal Serial Bus |
| **WAP** | Wireless Application Protocol |

# References

This section contains the locations of various specifications, document references and useful information where you can learn more about this subject.

[1] OMA-DRM-DRMREL-V2_0-20040130-D - Rights Expression Language V2.0

[2] OMA-Download-DRMCF-v1_0-20031113-C - DRM Content Format

[3] OMA-DRM-DCF-V2_0-200400614-D - DRM Content Format V2.0

[4] OMA-DRM-ARCH-V2_0-20040518-D - DRM Architecture

[5] OMA-RD_DRM-v2_0-20040420-C - OMA DRM Requirements

# Disclaimer

This document is based on Leliwa training materials.

Information in this document is subject to change without notice. Leliwa assumes no responsibility for any errors that may appear in this document.

This document may be freely redistributed. You can store it on any servers and make it available for public download. In such case it must be clearly indicated that it comes from Leliwa website www.leliwa.com

If you received only this file, you can download more Leliwa Technical Bulletins from the following address:

http://www.leliwa.com/downloads

If you want to be informed when the new bulletins are uploaded, please send a blank e-mail with Subject="Update_request" to bulletins@leliwa.com or click this link: bulletins@leliwa.com

**Leliwa Sp. z o.o.**

Plebiscytowa 1.122
PL-44-100 Gliwice
Poland
GPS: N50.2981°, E018.6561°

telephone: +48 32 376 63 05
fax: +48 32 376 63 07
Skype: leliwa_poland
email: info@leliwa.com

**Leliwa Telecom AB**

Orrpelsvägen 66
SE-167 66 BROMMA
Sweden
GPS: N59.3260°, E17.9464°

telephone: +46 8 4459430
email: info@leliwa.com