Digital Rights Management Version 1.0

Version 05-September-2002

Open Mobile Alliance
OMA-Download-DRM-v1_0-20020905-C

.

A list of errata and updates to this document is available from the Open Mobile Alliance™ Web site, http://www.openmobilealliance.org/, in the form of SIN documents, which are subject to revision or removal without notice.

This document is available online in PDF format at http://www.openmobilealliance.org/.

Known problems associated with this document are published at http://www.openmobilealliance.org/.

Comments regarding this document can be submitted to the Open Mobile Alliance™ in the manner published at http://www.openmobilealliance.org/technical.htm

Document History	
OMA-Download-DRM-v1_0-20020905-C	Current
OMA-Download-DRM-v1_0-20020905-p	Proposed

Contents

1. Scope

Open Mobile Alliance (OMA) Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the Open Mobile Alliance is to define a set of specifications to be used by service applications. The wireless market is growing very quickly, and reaching new customers and providing new services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of protocols in transport, session and application layers. For additional information on the WAP architecture, refer to "*Wireless Application Protocol Architecture Specification*" [WAPARCH].

The scope of OMA "*Digital Rights Management*" is to enable the controlled consumption of digital media objects by allowing content providers to express usage rights, e.g., the ability to preview DRM content, to prevent downloaded DRM content from being illegally forwarded (copied) to other users, and to enable superdistribution of DRM content. The defined technology is an initial DRM system that can be extended into a more comprehensive and secure DRM system.

2. References

2.1. Normative References

[CREQ] "Specification of WAP Conformance Requirements". WAP Forum™. WAP-221-CREQ. http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels". S. Bradner. March 1997. http://www.ietf.org/rfc/rfc2119.txt

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997. http://www.ietf.org/rfc/rfc2234.txt

[DRMREL] "DRM Rights Expression Language". Open Mobile Alliance™. OMA-Download-DRMREL-v1_0. http://www.openmobilealliance.org/

[DRMCF] "DRM Content Format". Open Mobile Alliance™. OMA-Download-DRMCF-v1_0. http://www.openmobilealliance.org/

[PUSHOTA] "Push OTA Protocol". WAP Forum™. WAP-235-PushOTA. http://www.openmobilealliance.org/

[RFC2046] "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types". N. Freed et al. November 1996. http://www.ietf.org/rfc/rfc2046.txt

[RFC2392] "Content-ID and Message-ID Uniform Resource Locators". E. Levinson. August 1998. http://www.ietf.org/rfc/rfc2392.txt

[RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1". R.Fielding et al. June 1999. http://www.ietf.org/rfc/rfc2616.txt

[WAE] "Wireless Application Environment Specification". WAP Forum™. WAP-236-WAESpec. http://www.openmobilealliance.org/

2.2. Informative References

[WAPARCH] "WAP Architecture". WAP Forum™. WAP-210-WAPArch. http://www.openmobilealliance.org/

[DLARCH] "Downloading Architecture and Overview". Open Mobile Alliance™. OMA-Download-ARCH-v1_0. http://www.openmobilealliance.org/

[DLOTA] "Generic Content Download Over The Air Specification". Open Mobile Alliance™. OMA-Download-OTA-v1_0. http://www.openmobilealliance.org/

[WSP] "Wireless Session Protocol". WAP Forum™. WAP-230-WSP. http://www.openmobilealliance.org/

[WDP] "Wireless Datagram Protocol". WAP Forum™. WAP-259-WDP. http://www.openmobilealliance.org/

[SDP] "SDP: Session Description Protocol". M. Handley, V. Jacobson. April 1998. http://www.ietf.org/rfc/rfc2327.txt

[SRTP] "The Secure Real Time Transport Protocol". M. Baugher et al. June 2002. http://search.ietf.org/internet-drafts/draft-ietf-avt-srtp-05.txt

3. Terminology and Conventions

3.1. Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2. Definitions

Asset	Content governed by rights. See DRM content.
Combined delivery	Delivery of the rights object and content together in a single message. See DRM message.
Composite object	A media object that contains one or more media objects by means of inclusion e.g. DRM messages, zip files.
Content	A media object
Consuming device	A mobile device consuming DRM content.
DRM agent	A user agent in the device that enforces the rights and controls the consumption of DRM content on the device.
DRM content	Content that is consumed according to a set of rights. DRM content may be in encrypted DRM Content Format or in plaintext delivered inside a DRM message
DRM message	A message containing a media object and an optional rights object. Media objects received inside a DRM message must not leave the device. The optional rights object defines additional consumption rules for the media object.
Forward-lock	A special case of combined delivery method where the DRM message includes only the media object and not a rights object at all. A set of default rights applies for the media object.
Media object	A digital resource e.g. a ringing tone, a screen saver, a Java game or a composite object.
Media type	A MIME media type.
Rights	Permissions and constraints defining under which circumstances access is granted to DRM content.
Rights issuer	An entity who issues rights objects.
Rights object	An instance of rights
Separate delivery	Delivery of the rights object and content via separate transports.
Superdistribution	A mechanism that (1) allows the end user to redistribute the encrypted DRM content to other end users through potentially insecure channels and (2) enables the receipients to obtain initial rights for the superdistributed DRM content.

3.3. Abbreviations

CEK	Content Encryption Key
DCF	DRM Content Format
DRM	Digital Rights Management
HTTP	Hypertext Transfer Protocol
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
REL	Rights Expression Language
WAP	Wireless Application Protocol
WSP	Wireless Session Protocol

4. Introduction

There is a need for content providers and operators to control the usage of downloaded media objects. Download is the means by which a media object is delivered to the device. Digital Rights Management (DRM) is the means to control the usage of the media object once it has been downloaded.

DRM enables content providers to define rules (rights) for how the media object should be used. It is possible to associate different rights with a single media object. Different rights may have different prices. A content provider can grant a user the rights to preview media objects for free and charge the user only for the full usage rights. Since the value lies in the rights and not in the media object itself, DRM makes it possible to sell the rights to use the media object, rather than selling the media object itself.

The rights can be delivered to the consuming device by downloading them together with the content or by sending the rights object separately from content (Figure 1.). The former case (combined delivery) is simpler whereas the latter case (separate delivery) provides more security by making it more difficult to steal the content. However, a complete DRM technology, including strong security between the consuming devices and content providers is not in the scope of OMA Download. The OMA Digital Rights Management follows common DRM practices taking into account the special requirements and characteristics of the mobile domain in order to support basic functionality with some level of security. This specification defines also a "forward-lock" special case of combined delivery where the DRM message does not contain a rights object. In that case a set of default rights apply for the media object.

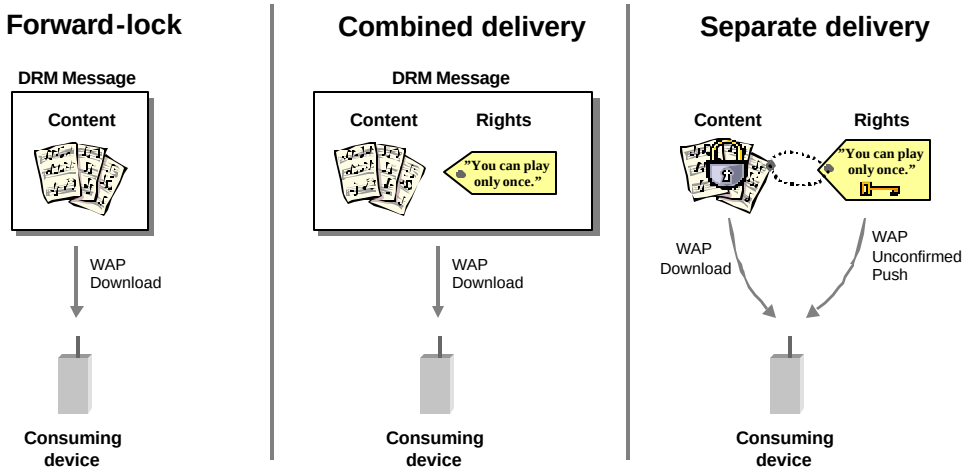

Figure 1. DRM methods

In the OMA Download "DRM scope", two problems are solved. The first problem is that there is currently no standardized way to prevent users from forwarding media objects from one device to another. The second problem is that for some business models there is no easy and convenient way for users to preview a media object before it is purchased. Users either pay before even having a chance to preview the media object, or they are presented with a "crippled" variant of the media object such as a low-quality version of a downloadable image or a small music fragment. For many media types, however, such "crippled " variants cannot easily be created (for example, executables). DRM also opens up new opportunities in a wide range of business models, e.g., content now can be assigned a lifetime, enabling subscription service providers to provide their customers content for a pre-determined time frame.

4.1. Forward-lock and combined delivery

For forward-lock and combined delivery content provider needs to package content, optionally with a rights object, into a DRM message. That message may be delivered to the device using e.g. the OMA Download mechanism. The format of the DRM message is defined in chapter 6. The consuming device renders the content according to the semantics

defined for the DRM message and the rights object that may have been included in the message. If a rights object is not included in the DRM message a set of default rights apply for the content as defined in section 5. It is up to the content provider to define what explicit rights are appropriate to use in each case. Typically the end user is charged for the combination of downloaded content and rights, but how that happens is out of the scope for this specification.

The figure below depicts a typical architecture of a combined delivery solution.

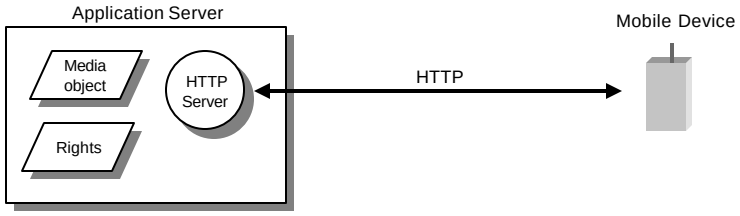

Figure 2. An example architecture for combined delivery

4.2. Separate delivery

In the separate delivery method the content provider needs to convert the plaintext media object into DRM content format (DCF) defined in the "*DRM Content Format*" specification [DRMCF]. This conversion includes symmetric encryption of the content making the DRM protected content object useless to parties not having access to the Content Encryption Key (CEK). Thus content in DRM format may be distributed via an insecure transport whereas a more secure transport (from DRM point of view) is used to deliver the rights object with the CEK as defined in chapter 5.5.

The figure below depicts a typical architecture of a separate delivery solution.

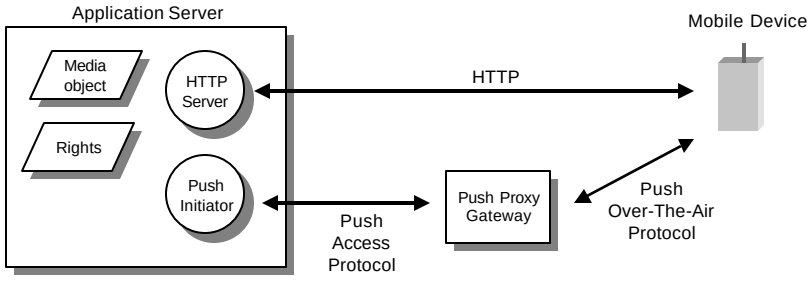

Figure 3. An example architecture for separate delivery

It is expected that pushing the rights in a GSM network is typically done by using unconfirmed push over connectionless session service like GSM SMS Profile of WDP. A WBXML encoded rights object and WAP Push headers fit in most cases into a single SMS message. Please note that since WAP Push is bearer independent SMS is only an example.

Using WAP push to deliver the rights object introduces some latency time between receiving the content and the rights. In order to enable a good user expirience this must be taken into account by both device manufacturers and service implementers. A mechanism that may be used by the service to indicate to the device that a rights object will be pushed is defined in this specification.

4.3. Superdistribution

The superdistribution case utilizes the flexibility provided by the separate delivery case to encourage sharing of media objects without compromising any business model behind the rights. The media object is allowed to pass from mobile device to mobile device through any channel, with the rights object being obtainable from the Rights Issuer via WAP push, as in the previous case. The location information for the Rights Issuer's application server is defined in the meta data of the DCF object. The mobile device contacts the Rights Issuer by opening a browsing session, which allows the user to choose the rights that s/he requires.

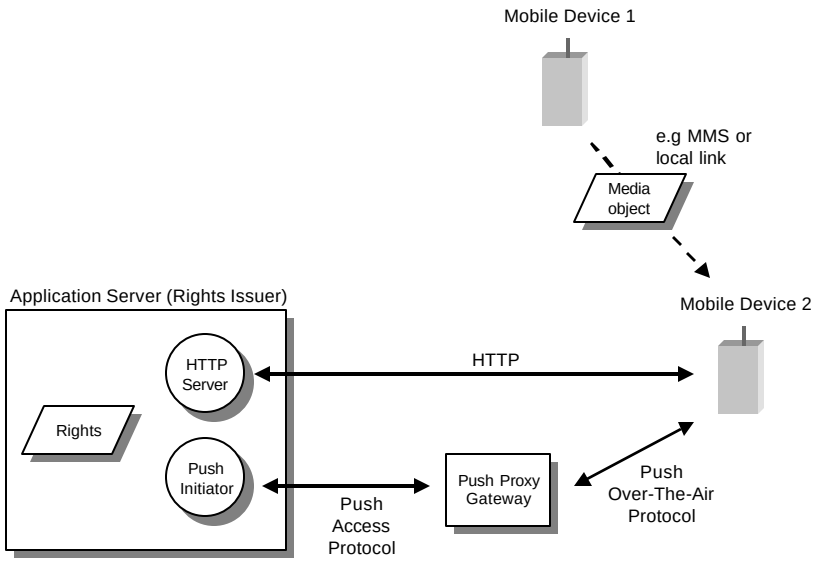

Figure 4 – An example architecture for Superdistribution

5. Digital Rights Management

5.1. Media Types

Three new media types are defined in OMA Digital Rights Management: rights, DRM content format and DRM message.

The media type, syntax and semantics for the rights object are defined in the "*Rights Expression Language*" specification [DRMREL].

The media type, syntax and semantics for the DRM content format are defined in the "*DRM Content Format*" specification [DRMCF].

The media type, syntax and semantics for the DRM message are defined in chapter 6 of this specification.

5.2. DRM Methods

This specification defines three DRM methods: forward-lock, combined delivery and separate delivery. The table below defines which media types must be supported by a device implementing a specific DRM method.

DRM method	Device must support media types	Definition
Forward-lock	application/vnd.oma.drm.*message*	Chapter 5.3
Combined delivery	application/vnd.oma.drm.*message* and application/vnd.oma.drm.*rights+xml*	Chapter 5.4
Separate delivery	application/vnd.oma.drm.*rights+xml*, application/vnd.oma.drm.*rights+wbxml* and application/vnd.oma.drm.*content*	Chapter 5.5

5.3. Forward-lock

The device MUST support the "forward-lock" method defined in this chapter.

In the forward-lock method the media object is wrapped into a DRM message and delivered to the device. The device is allowed to render the content but not to forward it to other devices.

The device MUST indicate support for the DRM message media type.

The device MUST support DRM messages that contain one media object. A device that supports only "forward-lock" MUST discard DRM messages that contain a rights object and SHOULD notify the user.

The device MUST NOT forward a media object that has been received in a DRM message. The device is allowed to play, display, execute and print the media object without any constraints. The device MUST NOT modify the media object.

5.4. Combined Delivery

The device MAY support the "combined delivery" method defined in this chapter. If the device supports the "combined delivery" method it MUST also support the "forward-lock" method.

In the combined delivery method a rights object and a media object is wrapped into a DRM message and delivered to the device, after which the device may render the content according to the rights object.

The device MUST indicate support for both DRM message media type and rights media type.

The device MUST support DRM messages that contain one rights object and one media object. In combined delivery the rights object and the media object are associated with each other by the DRM message. Since the association is external to the objects themselves the device MUST ensure that the rights information is preserved after the DRM message is received and possibly discarded.

The device MUST NOT forward a media object that has been received in a DRM message.

The device MUST NOT forward rights objects from the device.

The device MUST enforce the rights as defined in "Rights Expression Language" [DRMREL] when consuming the content.

The rights expression language governs the usage of content e.g. whether the media object is allowed to be rendered only once.

The rights expression language does not govern the local management of DRM content i.e the device MUST enable the end user to save, install, uninstall and delete DRM content.

The rights expression language does not govern the distribution of DRM content.

5.5. Separate Delivery

The device MAY support the "separate delivery" method and superdistribution defined in this chapter. If the device supports the "separate delivery" method it MUST also support the "combined delivery" and "forward-lock" methods.

In the separate delivery method the media object is always encrypted and converted into the DCF format. Typically the DCF object is downloaded to the device using e.g. OMA Download [DLOTA], after which the rights object is separately delivered to the device using WAP push. The service is expected to indicate this special behaviour to the device by using the mechanism defined in 5.5.1. After receiving the pushed rights object the device may render the media object. The WAP push should be targeted specifically for the DRM user agent.

The device is also allowed to forward (superdistribute) the DCF file to another device. However, rights objects are not allowed to be forwarded with the DCF i.e. the receiving device must acquire rights for the media object from the rights issuing service.

The device MUST indicate support for both rights media type and DRM content format media type (DCF).

In separate delivery the media object MUST be converted into the DRM content format.

The rights object is delivered to the consuming device using WAP push technology as defined in "Push OTA Protocol" specification [PUSHOTA]. The device MUST support unconfirmed push over connectionless session service using the Push OTA Protocol service primitive Po-Unit-Push. Other service primitives MAY also be supported.

The well known value for the Push Application ID of the DRM User Agent Push is

- URN: x-wap-application:drm.ua
- Number: 0x08

The device MUST be able to receive rights objects that are pushed to the DRM user agent using the Push Application ID defined above.

The device MUST NOT forward rights objects from the device.

The device SHOULD allow a DCF object to be forwarded from the device. Any transport mechanism supported by the device MAY be used.

The service may wrap the DCF object inside a DRM message without rights. In that case the device MUST NOT forward the DCF file from the device. Content inside a DCF can not be rendered without the key from the rights object. Thus the default rendering permissions defined in the forward-lock method do not apply for DCF objects delivered inside DRM message.

The device MUST enforce the rights as defined in "Rights Expression Language" [DRMREL] when consuming the content.

If there are no rights objects associated with a piece of DRM content the device MUST NOT consume the content. When attempting to consume such content, the device MUST give the end user the option to obtain rights from the rights issuing service by using the mechanism defined in 5.5.2. The exception to this is the case when the device knows that the rights object is currently being pushed to the device (chapter 5.5.1).If there are multiple rights objects associated with a piece of DRM content, each rights object MUST be treated individually, i.e. rights objects MUST NOT be combined. At any one time, there may be more than one rights object whose constraints are satisfied. When this is the case, the device MUST select one to enforce. This selection may be made automatically by the device based on some selection criteria, e.g. picking the least restrictive rights object, or it may be done based on user interaction. The selection process SHOULD favour the end user.

The rights expression language governs the usage of content e.g. whether the media object is allowed to be rendered only once.

The rights expression language does not govern the local management of DRM content i.e. the device MUST enable the end user to save, install, uninstall and delete DRM content. The rights expression language does not govern the distribution of DRM content.

5.5.1. Separate delivery indication

The usability of separate delivery is a challenge due to the potential latency time introduced by using WAP push to deliver the rights object to the device. This chapter defines a method that can be used to differentiate the separate delivery case from normal media object downloads.

When using HTTP for the delivery of the DCF file the service can indicate to the device that it intends to push the rights object separately to the device by adding a X-Oma-Drm-Separate-Delivery HTTP entity header to the HTTP response containing the DCF file. In addition to indicating that the rights object will be pushed, it is also possible to specify the expected time that it will take for the pushed rights object to arrive to the device.

```
X-Oma-Drm-Separate-Delivery : "X-Oma-Drm-Separate-Delivery" ":" [ delta-seconds ]
```

The optional delta-seconds parameter, defined in HTTP 1.1 specification [RFC2616], is an estimate how many seconds will pass before the rights object will arrive to the device. The value is only a rough estimate by the service.

The absence of the X-Oma-Drm-Separate-Delivery HTTP entity header indicates that no rights object will be pushed by the service.

The device MUST support the X-Oma-Drm-Separate-Delivery header.

5.5.2. Superdistribution

Superdistribution is facilitated (1) by allowing DCF formatted media objects to be forwarded from device to device and (2) by enabling devices to obtain rights for superdistributed content from the rights issuing service. The rights are obtained by opening a browsing session to the Rights-Issuer URL defined inside the DCF object. If the rights issuer decides to grant rights it pushes the rights object to the device.

The content provider is responsible of setting up a HTTP server that is able to respond to to requests made to the Rights-Issuer URL.

When obtaining rights for a DCF object, the device MUST make a HTTP GET request to the URL defined in the Rights-Issuer field of the DCF object. In order to facilitate the rights purchase dialog the device MUST support Wireless Application Environment as defined in [WAE].

When receiving a DCF formatted encrypted media object through means other than as described in [DLOTA] it is not known whether the content inside is suitable for the new device. Therefore it is necessary to check the suitability of the content prior to allowing the user to acquire new rights through the Rights-Issuer URL field.

The device MUST use the Content-Type field and the DataLen field to determine whether the content is suitable for the device before using the Rights-Issuer URL field. Where the device cannot determine that the content is appropriate for the device, the device SHOULD present a warning or error to the user, and SHOULD present a warning before navigating to the Rights-Issuer-URL

6. DRM Message Content Format

This section defines the content format for the DRM message that is used in the forward-lock and combined delivery methods. It is excpected that DRM message objects will not be stored persistently to the device.

6.1. DRM Message Media Type

The MIME media type for objects conforming to the format defined in this section MUST be

```
application/vnd.oma.drm.message
```

6.2. DRM Message Syntax

The DRM message syntax is based on a MIME multipart composite type in which one or more objects are combined in a single body. The following description is quoted from RFC 2046 chapter 5.1:

> *"In the case of multipart entities, in which one or more different sets of data are combined in a single body, a "multipart" media type field must appear in the entity's header. The body must then contain one or more body parts, each preceded by a boundary delimiter line, and the last one followed by a closing boundary delimiter line. After its boundary delimiter line, each body part then consists of a header area, a blank line, and a body area."*

The DRM message body MUST be according to the body of the multipart media type defined in RFC 2046 chapter 5 [RFC2046].

The DRM message MUST contain one or two body parts, one body part for each object. A DRM message MUST contain one media object that is not a rights object. In the forward-lock method, a DRM message MUST contain exactly one media object. In the combined delivery method, a DRM message MUST contain exactly one rights object and exactly one media object. If the rights object exists, it MUST be the first object in the DRM message. The rights object governs the consumption of the media object inside the DRM message. If the media object is internally a composite object, the rights MUST apply for each part of the composite object.

If HTTP or a MIME compliant protocol is used to transport the DRM message the boundary delimiter MUST be included as a parameter in the media type definition.

The boundary delimiter MUST NOT appear inside any of the encapsulated body parts, as described in the following quote from RFC 2046 chapter 5.1:

> *"As stated previously, each body part is preceded by a boundary delimiter line that contains the boundary delimiter. The boundary delimiter MUST NOT appear inside any of the encapsulated parts, on a line by itself or as the prefix of any line. This implies that it is crucial that the composing agent be able to choose and specify a unique boundary parameter value that does not contain the boundary parameter value of an enclosing multipart as a prefix."*

RFC 2046 defines a Content-Transfer-Encoding header that describes how a specific body part is encoded. The default value is 7bit encoding, as described in the following quote from RFC 2046 chapter 6.1:

> *"An encoding type of 7BIT requires that the body is already in a 7bit mail-ready representation. This is the default value -- that is, "Content-Transfer-Encoding: 7BIT" is assumed if the Content-Transfer-Encoding header field is not present."*

Content-Transfer-Encoding MUST only be used with body parts of DRM message, not with the whole body of the DRM message. The device MUST support the identity transfer encodings "7bit", "8bit" and "binary". Other non-identity Content-Transfer-Encodings like "base64" MAY also be supported.

A Content-ID header is used to associate a media object to a rights object. The conversion between the Content-ID header in the media object body part and the asset URI inside the rights object is done as defined in [RFC2392]. If the

media object is a DCF object the ContentURI field must match the Content-ID header. The device MUST support the Content-ID header.

7. Security Considerations (Informative)

7.1. Security trade-offs

There are obvious trade-offs in security due to the missing key management infrastructure. Reusing existing or emerging key management solutions, e.g. WIM/WPKI, is not directly possible for DRM purposes because the threat models in DRM and WIM are different. In WIM the end user is the target of attacks whereas in DRM the end user is the attacker himself.

Without the appropriate terminal key management infratructure in place there are no cryptographic means to protect against all security threats. Transport level security mechanisms, e.g. WTLS class 2, can be used to gain some security, but without terminal keys enabling device authentication and appropriate device security there will always be a possibility to use faked terminals to steal content. The use of WAP unconfirmed push over SMS for separate rights delivery, however, increases the security due to increased complexity of stealing the rights on the terminal.

The solution defined in this specification should not be considered to be a 100% secure DRM system, if in fact that is possible. Rather, it is a system that intends to keep the honest people honest. This should be reflected in the value of the content for which the OMA Digital Rights Management is used.

7.2. Device security

The consuming device controls how DRM content is used within the device. Since it is possible for the end user to install additional applications to some devices it is important in those cases to be able to differentiate authorized and unauthorized entities within the device as depicted in the figure below. Unauthorized entities are not likely to follow the DRM rules and restrictions and thus their access to plaintext DRM content should be prevented.

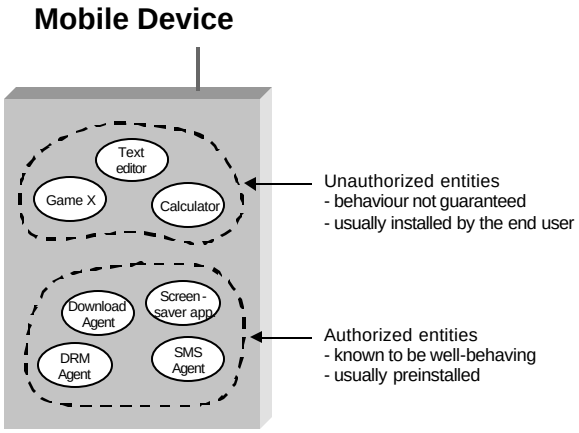

Figure 4. Authorized and unauthorized entities

This specification does not define the authorization mechanisms but sets some security requirements that imply that such a mechanism may be needed in some environments. However, devices that do not support installing new software do not need to implement any authorization mechanisms as the behaviour of all entities in the device can be guaranteed.

Unauthorized entities should not be able to access rights objects or DRM content in plaintext format. If the device has an open file system or removable memory that may imply local encryption of DRM content that has been received in plaintext format.

8. Handling of Streaming Media (Informative)

A media object governed by OMA DRM can be a digital resource of any kind (see definitions), including a description (meta-data) about other media object(s). For example, it can be a Session Description Protocol (SDP) record as a description of a media streaming session [SDP]. The OMA DRM mechanisms can then be used to indirectly control media objects via control of the meta-data. In this section we outline, as an example, how this can be used to enable rights management of streaming media.

The SDP record describing the streaming session is downloaded as the content in a DRM message. The media object description is, just like any other media object, recognized by its MIME type and processed by the responsible application. In the case of SDP the MIME type is application/sdp. The streaming player would then connect to the streaming server as specified in the session description, set up and receive the specified stream(s), and decode and render the streams as they arrive. With this mechanism, the OMA DRM mechanisms can be used to govern and control the use of media streams.

In order to allow for the same security level for streams as for downloaded objects, it is recommended that the streaming player is not allowed to store the media streams. Furthermore, the real-time media streams should be protected using a robust stream encryption mechanism suitable for a wireless environment, such as for example the Secure Real-Time Protocol [SRTP]. Without encryption of the streams, unauthorized parties would, with knowledge of the streaming server URL, be able to access the media streams. Consequently, an additional decryption key used to decrypt the streams must be conveyed to the UE. Such a decryption key for the stream can be conveyed within the attributes in the SDP record. The details of media stream encryption and key management for media streams are however outside the scope of this specification.

Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [CREQ].

A.1. Terminal Features

General terminal features

Item	Function	Reference	Status	Requirement
DRM-GEN-C-001	Forward-lock method	5.3	M	DRM-GEN-C-004 AND DRM-GEN-C-006
DRM-GEN-C-002	Combined delivery method	5.4	O	DRM-GEN-C-001 AND DRM-GEN-C-005 AND DRMREL: MCF
DRM-GEN-C-003	Separate delivery method	5.5	O	DRM-GEN-C-001 AND DRM-GEN-C-002 AND DRM-GEN-C-009 AND DRM-GEN-C-010 AND DRMREL: MCF DRMREL-GEN-C-001 AND DRMCF: MCF AND PushOTA: MCF
DRM-GEN-C-004	DRM message that contains one media object and no rights object	6	O	
DRM-GEN-C-005	DRM message that contains one media object and one rights object	6	O	
DRM-GEN-C-006	Identity Content-Transfer-Encodings “7bit”, “8bit” and “binary” for DRM message body parts	6	M	
DRM-GEN-C-007	“base64” Content-Transfer-Encoding for DRM message body parts	6	O	
DRM-GEN-C-008	Content-ID header in DRM message body parts	6	M	
DRM-GEN-C-009	Separate delivery indication	5.5.1	O	
DRM-GEN-C-010	Superdistribution	5.5.2	O	WAE: MCF

Appendix B. Examples (Informative)

B.1. Forward-lock Method

This example describes a forward-lock use case HTTP response that contains only one media object and no rights object. The device is allowed to view the image but not allowed to forward it to other devices.

```
HTTP/1.1 200 OK
Content-type: application/vnd.oma.drm.message;
              boundary=boundary-1
Content-Length: 574

--boundary-1
Content-type: image/jpeg
Content-Transfer-Encoding: binary

...jpeg image in binary format...
--boundary-1--
```

B.2. Combined Delivery

This example describes a combined delivery use case HTTP response that has the DRM message in the HTTP body. The DRM message contains one rights object and one jpeg media object. The device is allowed to preview the image once but not allowed to forward it to other devices.

```
HTTP/1.1 200 OK
Content-type: application/vnd.oma.drm.message;
              boundary=boundary-1
Content-Length: 893

--boundary-1
Content-type: application/vnd.oma.drm.rights+xml
Content-Transfer-Encoding: binary

<o-ex:rights
   xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
   xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
>
   <o-ex:context>
      <o-dd:version>1.0</o-dd:version>
   </o-ex:context>
   <o-ex:agreement>
      <o-ex:asset>
         <o-ex:context>
            <o-dd:uid>cid:4567829547@foo.bar</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:display/>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>

--boundary-1
Content-type: image/jpeg
Content-ID: <45678929547@foo.bar>
Content-Transfer-Encoding: binary

...jpeg image in binary format...
--boundary-1--
```

B.3. Separate Delivery

This example describes the first step of the separate delivery where the DCF file is delivered using HTTP. The service expects the WAP Push of the rights object to be delivered in 12 seconds and indicates that to the device by using the X-Oma-Drm-Separate-Delivery HTTP header.

```
HTTP/1.1 200 OK
Content-type: application/vnd.oma.drm.content;
Content-Length: 1234
X-Oma-Drm-Separate-Delivery: 12

...DRM content in DCF format...
```

Appendix C. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	05-September-2002		The initial version of this document.