Content protection
AV over IP

Draft A
19 August 2010

Confidentiality and Intellectual Property Notice

The contents of this document are strictly confidential and have been made available to you as a member of the DTG. You are prohibited from distributing, circulating or otherwise sharing this document with any individual, group or company that is not a member of the DTG. The BBC reserves all right, title and interest in this document, its content and subject matter. If you give the BBC suggestions, comments and other feedback you agree that the BBC may freely use, disclose, reproduce, licence and distribute such feedback as it sees fit.

Table of contents

1 Introduction

This document contains the detailed specification for the content protection concepts introduced in the white paper, Canvas Content Protection for Online Video Content[1].

A range of content protection options are specified. The options vary both in complexity for the content provider and in the level of protection provided. Content providers may use any of the supported options according to their needs.

Section 2 describes a set of content protection options to be provided by the core media playback function of the device.

In addition, specific presentation environments may provide additional mechanisms for content protection. Where appropriate, these are also supported, to maintain content compatibility. These are described in sections 3, 4 and 5.

The table below summarises the features and capabilities of the different content protection options.

Protection mechanism	Authentication	Content encrypted in delivery	Content encrypted on server	Support for HTTP caches	Hardware decryption	Configurable output controls	Offline playback of downloads	Streaming
§2.1 – Simple HTTP	N	N	N	Y	N/A	N	N	Y
§2.2 – Authentication only	Y	N	N	Y	N/A	N	N	Y
§2.3 – Authentication only using MS3	Y	N	N	Y	N/A	Y	N	Y
§2.4 – TLS streaming	Y	Y	N	N	N	N	N	Y
§2.5 – Marlin MS3 secure content delivery	Y	Y	Y	Y	Y	Y	N	Y
§2.6 – Marlin Broadband DRM	Y	Y	Y	Y	Y	Y	Y	Y
§3.1 – Flash HTTP	N	N	N	Y	N/A	N	N	Y
§3.2 – Flash RTMP	Optional	N	N	N	N/A	N	N	Y
§3.3 – Flash RTMPe	Optional	Y	N	N	N	N	N	Y
§4 – MHEG IC without encryption	N	N	N	Y	N/A	N	N	Y
§4 – MHEG IC with encryption	N	Y	Y	Y	Y	N	N	Y

A number of sequence diagrams are shown in this document. These provide a simplified view of the interactions that take place with each mechanism.

To illustrate the interactions between applications and the underlying media playback functions of the device, a component labelled 'MediaRouter' is shown to which represents the application's interface onto the media playback implementation.

[1] http://www.projectcanvas.info/index.cfm/technology/content-protection/

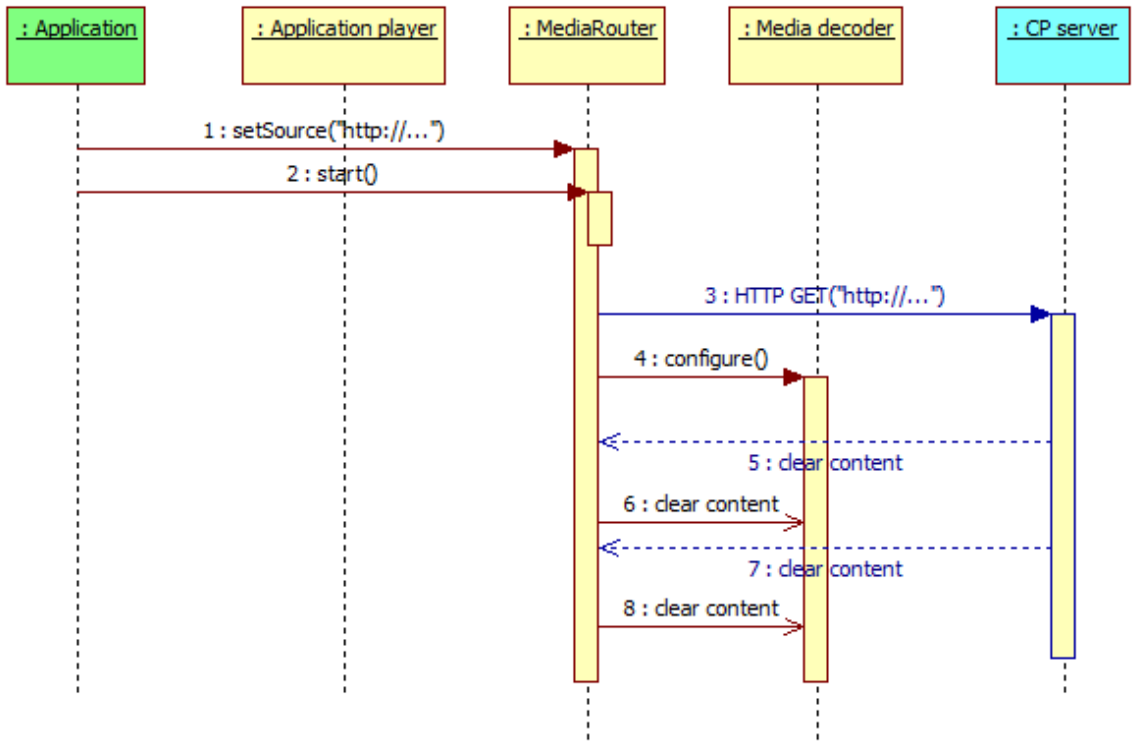
2 A/V delivery using Canvas components

2.1 No protection

2.1.1 Description

The simplest form of content delivery provides no protection. The Canvas device is not authenticated and there is no encryption of the content.

In this scheme, a direct reference to the media is passed by the application. For example, for content streamed over HTTP, the high level interactions would be as follows:



Note: interactions between the MediaRouter and Media decoder objects shown are illustrative only.

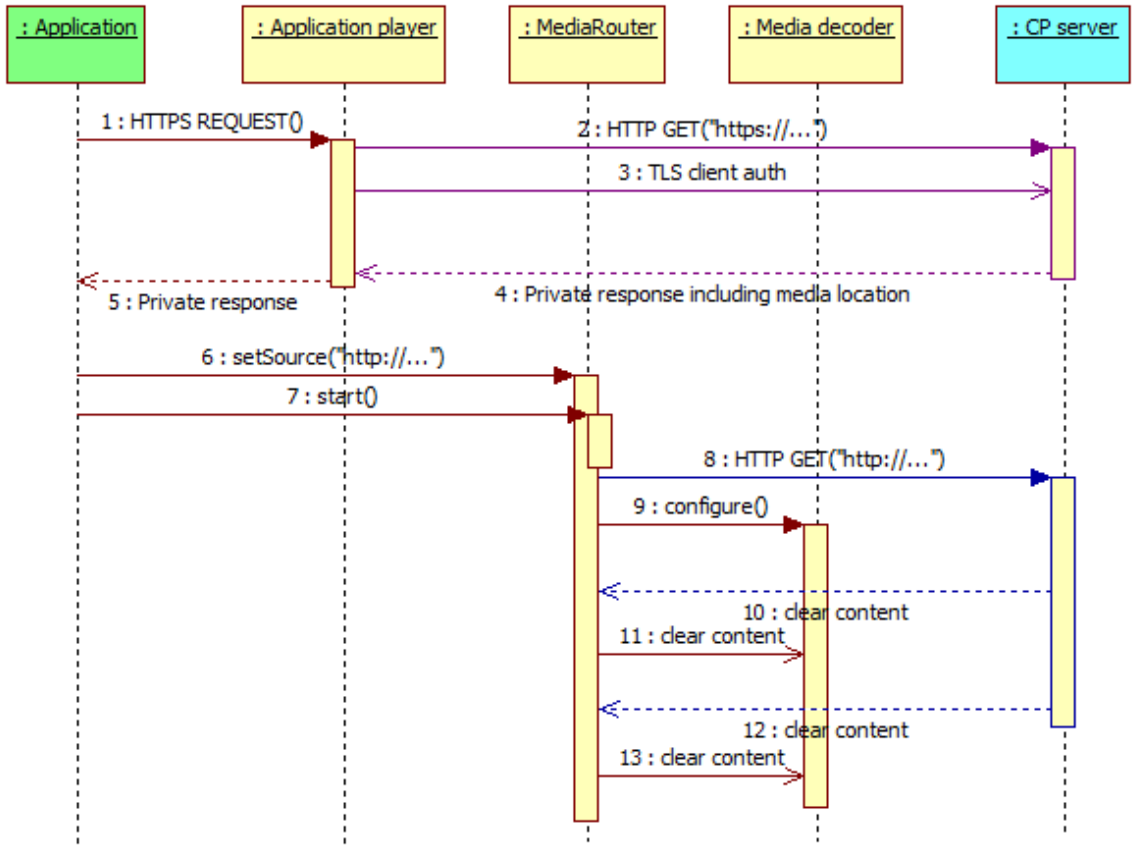
This content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

2.2 Simple device authentication

2.2.1 Description

This mechanism provides authentication of the Canvas device to the content provider for an interaction which takes place prior to the start of media playback between an application running on the Canvas device and a content provider's server. This mechanism can be used to deliver a media URL securely into the device for use in the same way as described in section 2.1.

When the media URL is subsequently used to request the content, the request will be made in the clear so the content URL could be captured. Content providers can use techniques such as time-limited tokens to limit the usefulness of URLs captured in this way. Note: one-time URLs cannot be used with this mechanism because the URL may be used more than once if the player needs to seek within the stream.



2.2.2 Detailed specification

In this mechanism, the secure interaction with the content provider's server uses the capabilities of the application player to set up a secure connection. The mechanism for this will vary depending on the particular application player or presentation engine being used.

The player makes a TLS connection to the server requested by the application and performs client authentication if requested by the server. The private key and certificate chain specified in section 2.9 shall be used.

Since the media playback element of this mechanism is exactly the same as the basic case of no protection, this content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

2.2.3 Compliance and robustness requirements

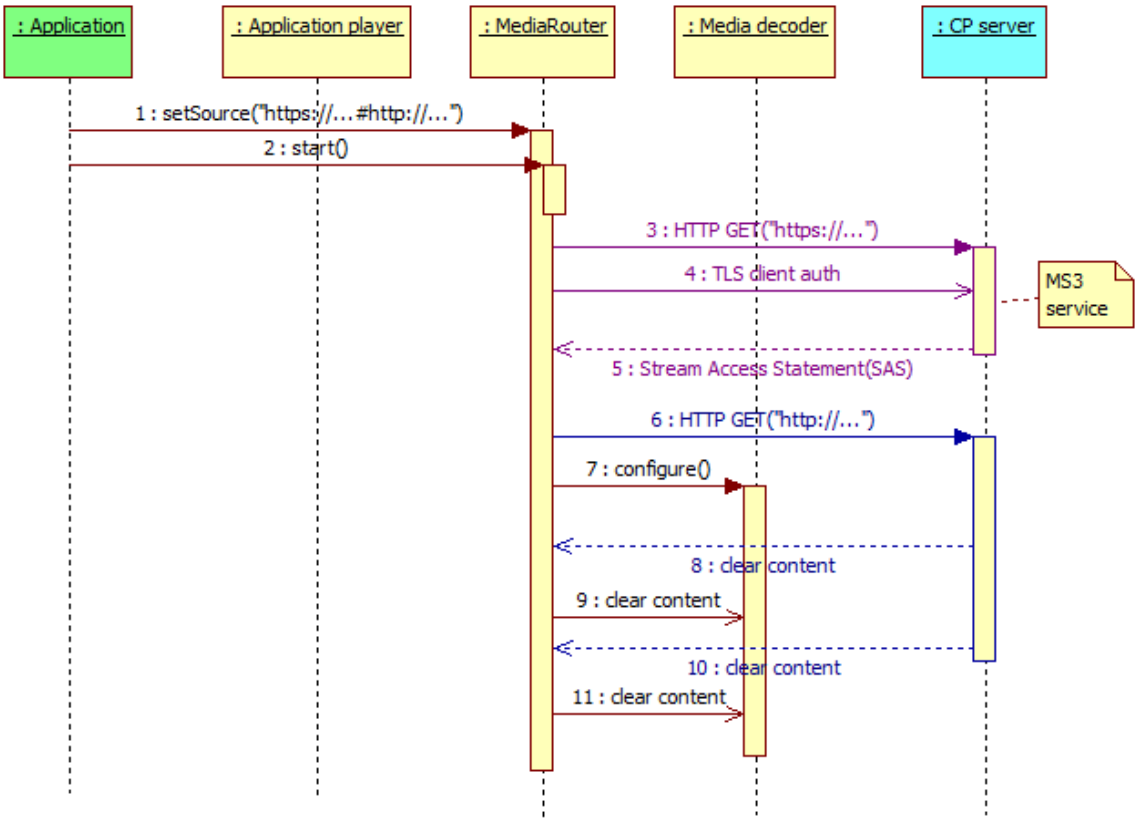
When this mechanism is used the private key for client authentication shall be protected to the same degree as required by the Marlin Robustness rules[1].

2.3 Device authentication using MS3

2.3.1 Description

This mechanism provides authentication of the Canvas device to the content provider, allowing the content provider to reveal parts of the content's location only to genuine Canvas devices. It does not include encryption of the content itself. In addition, the request made to obtain the content will be sent in the clear so the content URL could be captured. Content providers can use techniques such as one-time or time-limited tokens to limit the usefulness of URLs captured in this way.

[1] Part of the Marlin Client Adopter agreement, available from http://www.marlin-trust.com/downloads/agreement



Note: interactions between the MediaRouter and Media decoder objects shown are illustrative only.

2.3.2 Detailed specification

In this mechanism, a compound URI is passed by the application to the MediaRouter interface. The compound URI includes an https: URL for an MS3 service (the S-URL), together with a URI template for an unencrypted delivery mechanism with which to obtain the content (the C-URIT). The format of the compound URI is specified in section 3.4.2 of the Marlin Simple Secure Streaming (MS3) specification version 1.0[1].

When presented with a URI of this type, the implementation behind the MediaRouter interface shall behave as described by the MS3 specification. In summary:

1. The implementation first makes a secure connection to the MS3 service with client authentication using the private key and certificate chain specified in section 2.9.
2. The server responds with a Stream Access Statement (SAS) which can include an authenticator element to be substituted into the media URL and which also includes a set of output control requirements to be met whilst the content is being played.
3. The implementation forms a URL for the content (the C-URL) using the C-URIT and the authenticator.
4. The content is streamed using the C-URL as if it had been provided directly by the application.
5. For the period that the content is being presented, the output control mechanisms shall be configured to satisfy the output control requirements provided by the SAS within the context of section 2.8.

[1] Marlin specifications are available from http://www.marlin-community.com/develop/downloads/specifications

When a request is made to seek within the stream, the implementation shall first make one attempt to use the existing C-URL for the new request. If this fails, the implementation shall return to the MS3 service for a new SAS.

2.3.3 Compliance and robustness requirements

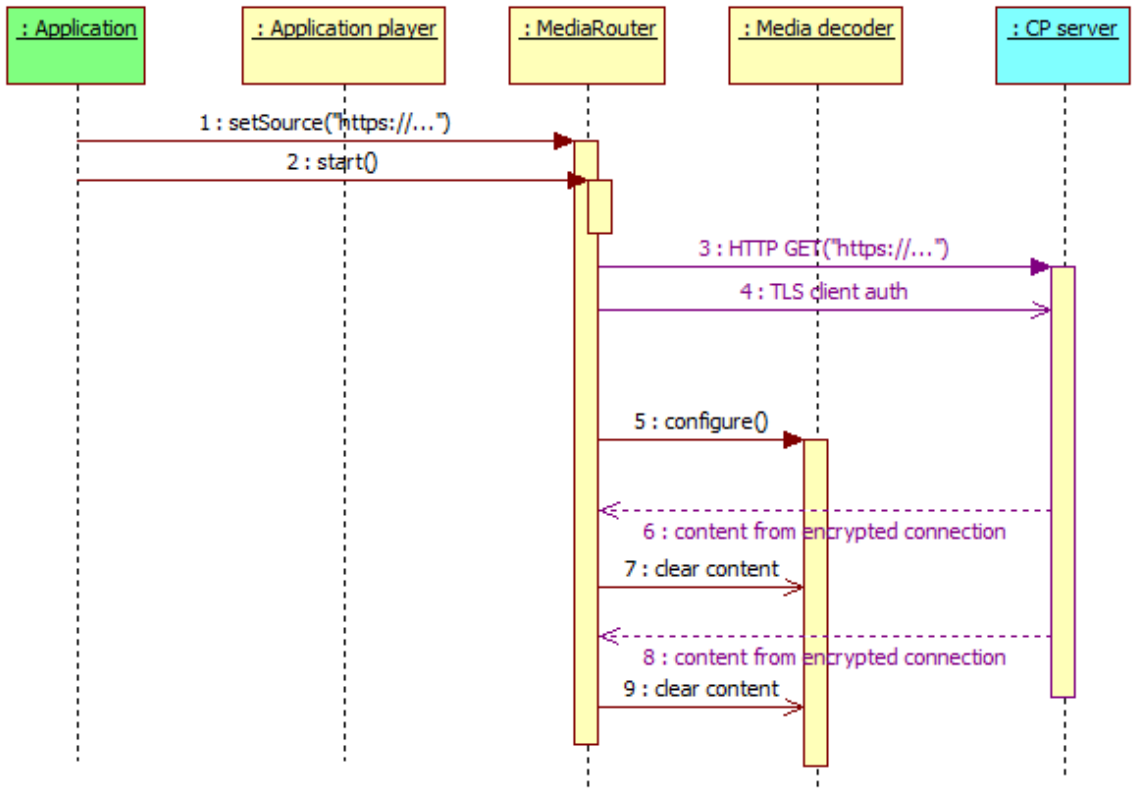
The Marlin Compliance and Robustness rules shall apply to the implementation of this mechanism. Note: as content is not encrypted in this delivery mechanism, the applicable requirements here relate primarily to the 'do not store' flag, the output restrictions and the handling of the TLS client certificate.

2.4 Device authentication and transport encryption

2.4.1 Description

This mechanism allows content to be delivered through an encrypted transport using TLS. In addition, the Canvas device is authenticated to the content provider. The content does not appear in the clear in the home network but would be stored unencrypted on the content provider's servers.

Implementations are not required to support, and content providers should not attempt to deliver, content at bitrates exceeding 2.5 Mbps (for AES-128 encryption) or 4 Mbps (for RC4 encryption) using this method. It is expected that on many devices, software-only implementations will be able to meet this requirement.



Note: interactions between the MediaRouter and Media decoder objects shown are illustrative only.

2.4.2 Detailed specification

In this mechanism, an https: URL is passed by the application to the MediaRouter interface. The device makes a TLS connection to the server specified in the URL and performs client authentication if requested by the server. The private key and certificate chain specified in section 2.9 shall be used.

Once the secure connection has been established, the implementation shall proceed from then on as if a simple http: URL had been used.

This content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

2.4.3 Compliance and robustness requirements

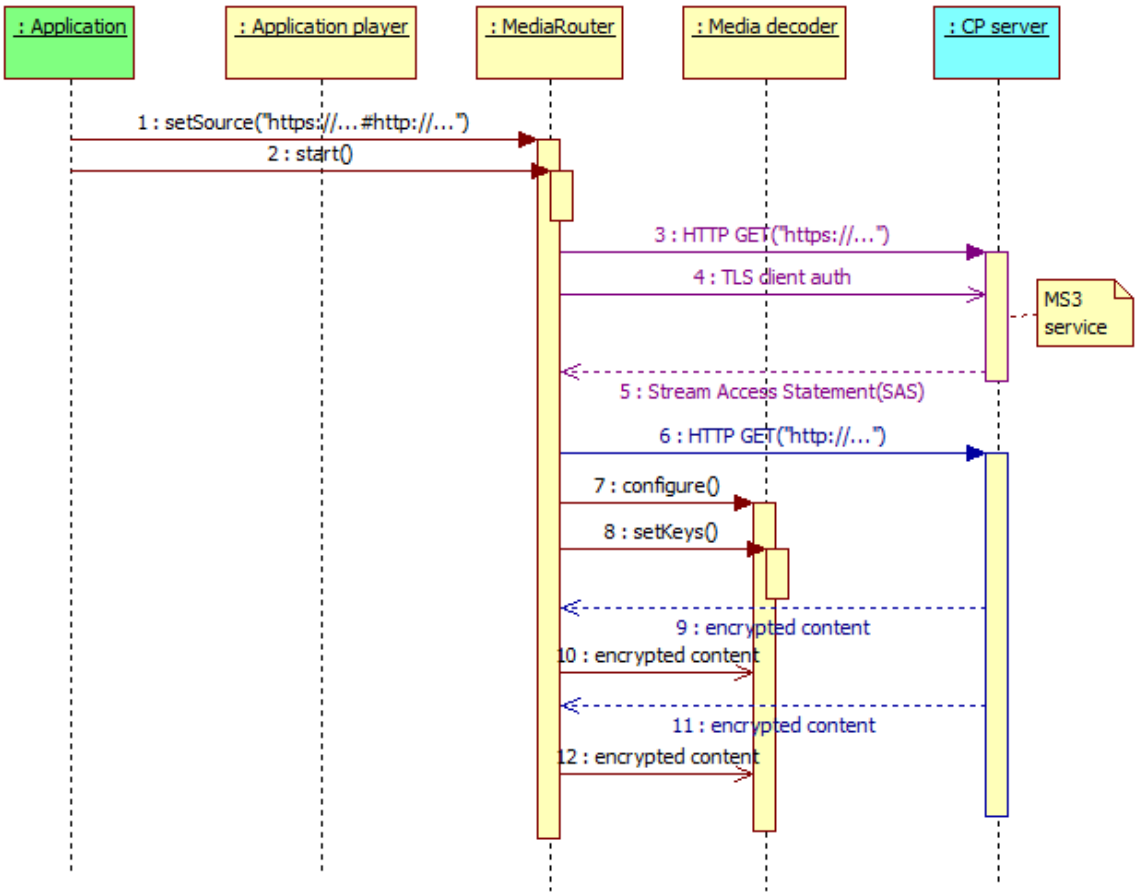
For content delivered in this manner:

- Devices shall not provide any means for the user to export the content from the device.
- Content shall not be stored on the device other than as required temporarily for buffering purposes.
- The private key for client authentication shall be protected to the same degree as required by the Marlin Robustness rules.
- No robustness rules apply to the handling of session keys or decrypted content with this scheme. As such, this mechanism aims to provide secure delivery of content into the device but provides limited protection against a determined attacker able to tamper with the device.

2.5 Device authentication and encrypted content delivery

2.5.1 Description

This mechanism allows encrypted content to be delivered, with the keys and output control information being provided through a secure, authenticated connection. The content does not appear in the clear in the home network or on the content provider's servers.



Note: interactions between the MediaRouter and Media decoder objects shown are illustrative only.

2.5.2 Detailed specification

In this mechanism, a compound URI is passed by the application to the MediaRouter interface. The compound URI includes an https: URL for an MS3 service (the S-URL), together with a URI template for an unencrypted delivery mechanism with which to obtain the content (the C-URIT). The format of the compound URI is specified in section 3.4.2 of the Marlin Simple Secure Streaming specification version 1.0.

When presented with a URI of this type, the implementation behind the MediaRouter interface shall behave as described by the MS3 specification. In summary:

1. The implementation first makes a secure connection to the MS3 service with client authentication using the private key and certificate chain specified in section 2.9.
2. The server responds with a Stream Access Statement (SAS) which can include an authenticator element to be substituted into the media URL and which also includes a set of output controls to be applied whilst the content is being played.
3. The implementation forms a URL for the content (the C-URL) using the C-URIT and the authenticator.
4. The content is streamed using the C-URL as if it had been provided directly by the application.
5. Before the content is decoded, the media decryption function is configured to decrypt the content using the keys obtained from the SAS.
6. For the period that the content is being presented, the output control mechanisms shall be configured to satisfy the output control requirements provided by the SAS within the context of section 2.8.

Note: steps 3 and 4 can proceed in parallel with steps 1 and 2 if C-URIT does not require an authenticator. This will result in improved start up times.

When a request is made to seek within the stream, the implementation shall first make one attempt to use the existing C-URL for the new request. If this fails, the implementation shall return to the MS3 service for a new SAS.

The C-URL may reference any supported streaming protocol specified in the IP Content Delivery specification, including multicast delivery.

Devices shall provide hardware-accelerated decryption for content delivered in this manner. Bitrates up to the maximum required for unencrypted content shall be supported.

Note: any additional functionality required in order to support protected linear IP channels will be added in a future revision of this specification.

2.5.3 Compliance and robustness requirements

The Marlin Compliance and Robustness rules shall apply to the implementation of this mechanism.

2.6 DRM protected content

2.6.1 Description

This mechanism uses the functions of the Marlin Broadband DRM system to protect the content covering use cases such as:

- Content downloaded for playback at a later time
- Multicast IP linear channels requiring stored licences

The DRM system can also be used for streaming use cases if desired.

Two modes of licence acquisition are supported:

1. Licence acquisition prior to playback, initiated as a result of interactions between an application running on the device and an external ‘web store’. The location of the web store

may be known to the application (in the case of a VOD store app) or may be obtained from information held in the metadata system (in the case of IP linear channels).

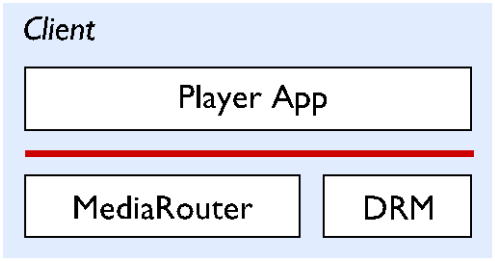
2. Licence acquisition on demand, triggered by an attempt to play protected content for which the device does not already have a valid licence.

For the period that the DRM-protected content is being presented, the output control mechanisms shall be configured to satisfy the output control requirements specified in the DRM licence within the context of section 2.8.

Implementations shall comply with the Marlin Broadband Delivery System Specification v1.2. Implementations shall provide a Full Implementation as defined by the Marlin Broadband Network Service (BNS) specification v1.2 with support for BNS Extended Topologies. Devices shall indicate their support for BNS Profile and BNS Extended Topologies in the manner described in section 6 of the BNS specification.

2.6.2 Client architecture

The diagram below shows a highly simplified view of the client device (a more detailed view is provided in section 2.7).

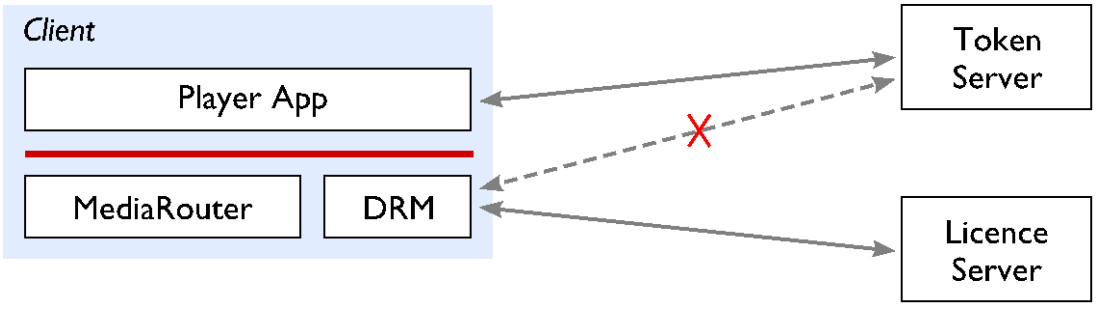


There are three components:

- Player App. This is the user-facing application through which the user selects content to watch or download and through which payments are made. This will generally be a service specific application but could be the UI in the case of IP linear channels.
- DRM. This is the system component that encapsulates the licence acquisition functions of the DRM system, providing a high level interface to the application. It stores and manages licences.
- MediaRouter. This is the interface onto the components that play back media content.

2.6.3 Licence acquisition

In the Marlin DRM system, licence acquisition is initiated via an 'action token' which is acquired from a web service. For Canvas, action tokens are only acquired by applications and there is no provision for token acquisition by the lower level components directly, as shown in the following diagram:



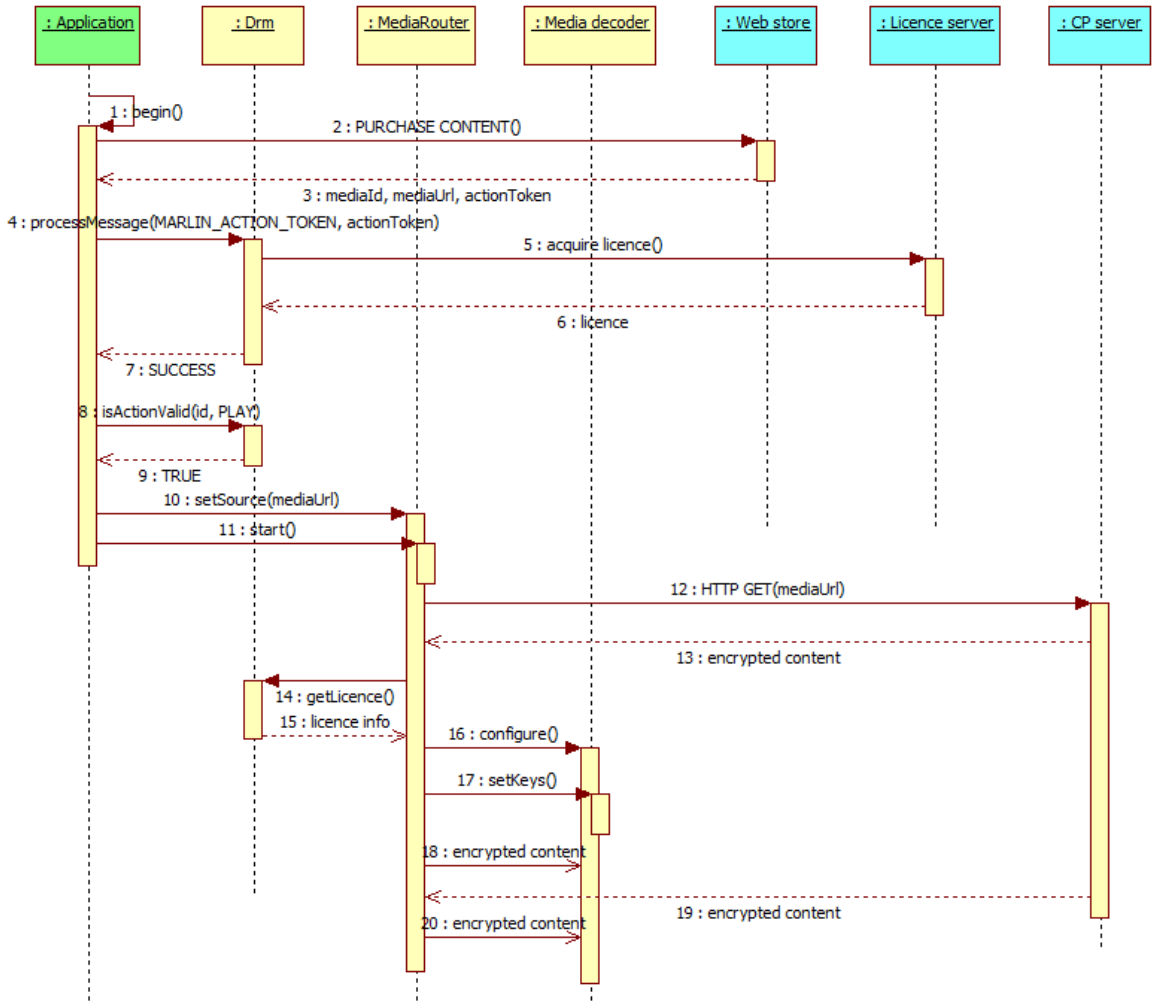
Note: any additional functionality required in order to support protected linear IP channels will be added in a future revision of this specification.

2.6.4 Example usage

The following sequence diagram shows a simple example using Marlin to protect a VOD streaming session. In this case, licence acquisition is being initiated in advance of playback. A similar process would be used for downloaded content.

Firstly, the application performs a transaction with the web store and obtains an action token. This is passed through the Drm interface where the DRM client acquires a licence from the licence server. After this process is complete, the application may optionally make an advance check that the DRM system now has a licence that will permit playback.

When playback is requested, the implementation underlying the MediaRouter and Drm interfaces obtains the information required and begins decoding the media.



Note: interactions between the MediaRouter and Media decoder objects, and between the Drm object and the MediaRouter are illustrative only. These interactions are implementation dependent.

2.6.5 DRM API

The DRM functions are provided to applications through an API which is not specific to Marlin but which could provide an interface to a number of DRM systems. It provides two APIs:

processMessage (messageType: Integer, message: String, out result: Integer) – requests that the DRM system process a message. In the case of Marlin, the supported message type is an Action Token.

isActionValid (drmMediaIdentifier: String, action: Integer, out result: Boolean) – tests whether a particular action (play, export, move etc.) would, at the time of the call, be allowed.

In addition, applications are informed via an event when an attempt is made to play back encrypted content for which no valid licence exists on the device. This event includes the Content ID and Rights Issuer URL obtained from the content.

2.6.6 Management of licences

Devices shall be able to store DRM licences. Licences are required for three types of content:

- Downloaded content stored on the device
- Recordings made from protected linear IP channels and stored on the device
- Content not stored on the device (e.g. content to be streamed or downloaded in future)

Devices shall check all stored licences at least once every 24 hours against the following conditions. If neither condition 1 nor condition 2 applies to a licence, it shall be retained, otherwise it shall be deleted.

1. the licence has a 'urn:marlin:core:node:attribute:expiration-date' attribute that represents a date in the past, or
2. the licence satisfies all of the following conditions:
    - it is not required by any content stored on the device, and
    - it has a 'urn:marlin:core:node:attribute:expiration-date' attribute that is more than 30 days in the future or has no such attribute, and
    - it was acquired more than 30 days ago

Note: this ensures that where a licence relates to content that is stored on the device, the licence will not be discarded until the content is deleted or the licence expires.

It is implementation dependent whether licences associated with content stored on the device are stored with the media or in a separate database.

Implementations must take account of the following scenarios:

- Downloads may reference multiple content IDs (e.g. for different tracks) but would normally have a single licence covering them.
- A recording may reference multiple content IDs and multiple licences (e.g. for different temporal parts of the recording).
- Multiple recordings may require the same licence or licences (e.g. if they were made from the same IP channel on the same day).

Note: this means that to store licences with the content, licences may have to be copied as they may relate to multiple assets stored on the device.

Where content formats allow DRM licences to be embedded and delivered to the device within the media, devices shall search any such licences first, prior to searching elsewhere to find a valid licence for a requested operation.

2.6.7 Compliance and robustness requirements

The Marlin Compliance and Robustness rules shall apply to the implementation of this mechanism.

2.6.8 Personalisation

Marlin clients must be 'personalised' in order to interact with Marlin licence servers and to handle Marlin licences.

Devices shall implement a mechanism that ensures that the device is personalised before any third party application runs for the first time. Any requirement to access a remote server to perform this task shall be included as part of the device's software upgrade procedure.

2.7 Device architecture for content protection and DRM

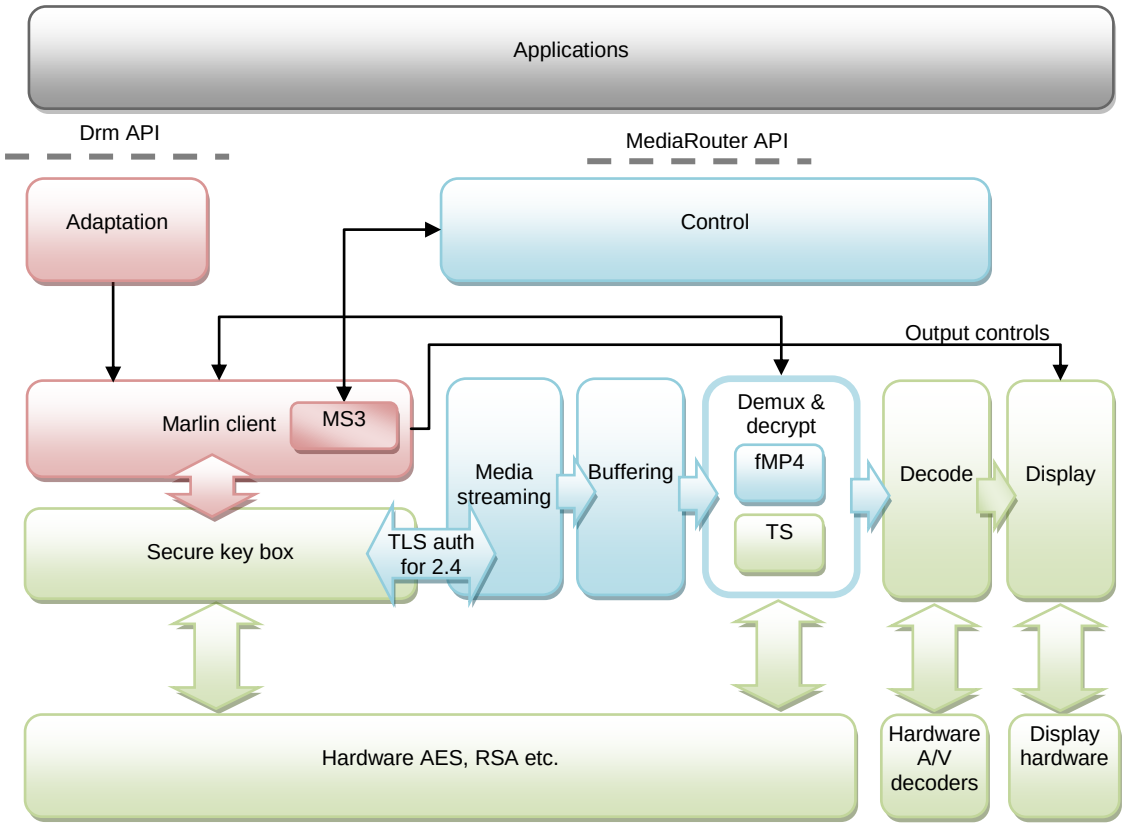
The following figure shows an architectural model of the content protection and playback functions of the device. The components shown in red are expected to be provided by the DRM provider. Components in green are lower-level hardware or firmware components, typically specific to the particular SoC being used. The remaining components are shown in blue.

The Marlin DRM client is accessed by applications through the 'Drm' API. The media playback functions are accessed through the 'MediaRouter' API.

The media playback control function interacts with the MS3 client in order to obtain any 'authenticator' required to access content. See section 2.3.

The Marlin DRM client uses secure key management functions to protect its secrets.

The media streaming function also requires secure key management capabilities to implement client authentication for TLS streaming. See section 2.4.



2.8 Output controls

Devices shall support the following output control mechanisms:

- HDCP
- CGMS-A
- Disabling of analogue outputs

The set of output controls in force at any point in time shall be sufficient to meet the requirements of all A/V presentation sessions that are active. The set of output controls shall be evaluated each time presentation of content begins or ends.

In accordance with D-Book 6.2.1, HDCP shall be applied at all times unless the user has specifically configured the device to apply HDCP only where explicitly signalled. Note: this is due to the long periods of black picture that result when the HDCP state is changed.

For all other output control mechanisms, the mechanism shall not be applied unless it is explicitly required for content that is currently being presented.

Where a content licence imposes constraints on analogue outputs that the device does not support, the analogue outputs shall be disabled.

When disabling the analogue video outputs, devices shall act as follows:

- If a digital video output is active, the analogue output shall be disabled with no other feedback provided.
- If no digital display device is active, the video shall not be presented on the video plane and an on-screen dialogue shall be presented to the viewer.

Note: the Consumer Device Platform specification requires that the device has no analogue HD outputs.

2.9 Client certificates

The following table specifies the TLS client certificates that shall be used for the protection mechanisms described in this specification.

Protection mechanism	Certificate to be used	Reference
Simple device authentication (section 2.2)	OEMDEVICE_AUTH. Certificate specific to the device model but not specific to an individual device.	Platform Trust Model.
Device authentication using MS3 (section 2.3)	Marlin Nemo Signing Certificate. This certificate is specific to the individual device.	Marlin Core System Specification §9.4.1.
Device authentication and transport encryption (section 2.4)	OEMDEVICE_AUTH. Certificate specific to the device model but not specific to an individual device.	Platform Trust Model.
Device authentication and encrypted content delivery (section 2.5)	Marlin Nemo Signing Certificate. This certificate is specific to the individual device.	Marlin Core System Specification §9.4.1.

2.10 Protected content formats

2.10.1 MPEG-2 transport stream

As defined in the IP Content Delivery specification, the encrypted content format for MPEG2-TS content is IEC 62455. The Marlin Broadband Transport Stream format specifies how Marlin-specific information is carried in an IEC 62455 compliant transport stream.

Encrypted streams intended for use with MS3 (see section 2.5) or Marlin Broadband (see section 2.6) shall comply with the requirements of the Marlin Broadband Transport Stream format specification version 1.0.2. Devices are not required to support Silent Rights or Preview Rights URL signalling.

2.10.2 MP4

Proposed Canvas shareholders are involved in industry standardisation activities including DTG, OIPF, 3GPP, DECE and MPEG and take the work of these groups into account when deciding on content formats. These standardisation activities still have some way to go in agreeing a common approach to MP4 encryption, particularly where support for both progressive download and adaptive bitrate streaming is required. This specification specifies an encrypted MP4 format for progressive download only. A future revision of this specification will address the requirement for an encrypted, fragmented MP4 file format.

Devices shall support MP4 content encrypted using the OMA PDCF format.

Encrypted streams intended for use with MS3 (see section 2.5) or Marlin Broadband (see section 2.6) shall comply with the requirements of the OMArlin Specification version 1.0.3. Devices are not required to support Silent Rights or Preview Rights URL signalling. Devices shall support embedded licences.

3 Presentation engine specific content protection: Flash

Some presentation environments provide additional A/V content delivery mechanisms. Detailed specification of these delivery mechanisms is outside the scope of this document.

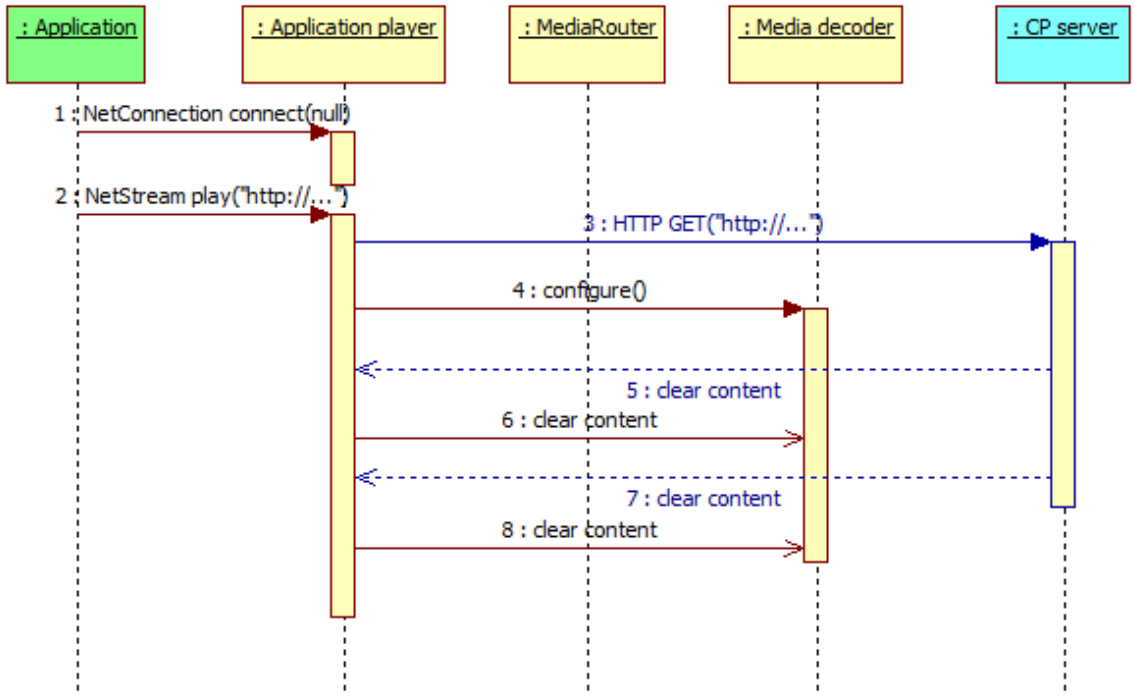
Consideration of the use of Flash as a presentation technology is ongoing. As part of this we are looking at the content protection aspects of Flash-specific delivery technology. The following sections provide a high level description of these. These delivery mechanisms would be available to Flash applications only.

3.1 Flash HTTP streaming

3.1.1 Description

This mechanism involves using the A/V streaming capabilities of the Flash player.

Simple HTTP streaming offers no specific content protection. It is the equivalent of the mechanism in section 2.1. There is no encryption and no authentication of the client.



3.1.2 Implementation requirements

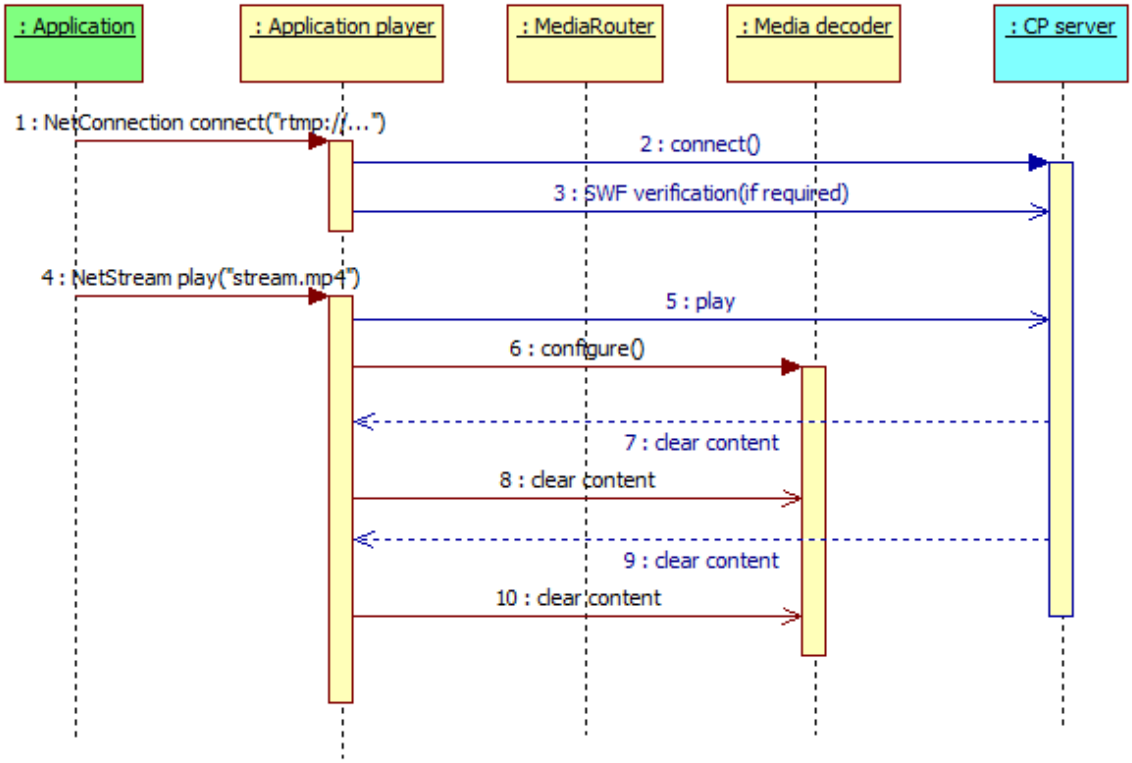
This content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

3.2 Flash RTMP streaming

3.2.1 Description

This mechanism does not encrypt the content. However, it uses a protocol for content delivery that makes it more difficult to capture content streamed over it.

A simple form of authentication is possible using SWF verification which is intended to ensure that the content is only delivered to the content provider's player. This makes acquisition of the stream more difficult for an attacker. RTMP itself does not offer *device* authentication.



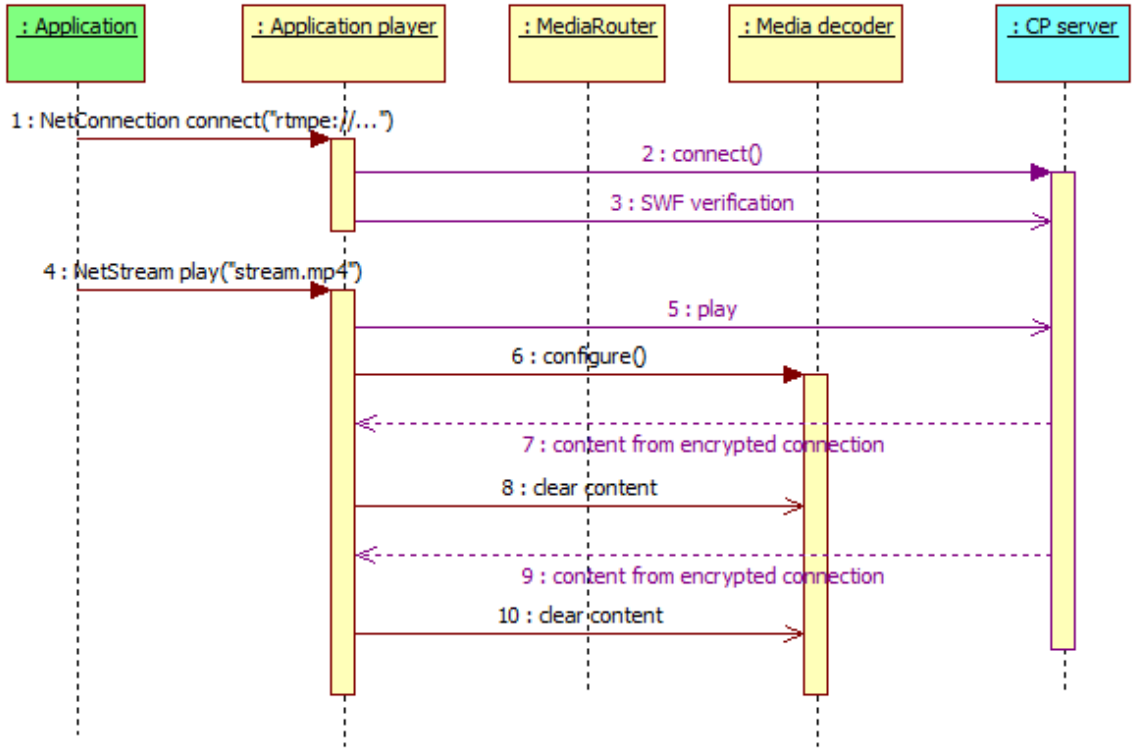
3.2.2 Implementation requirements

This content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

3.3 Flash RTMPe streaming

3.3.1 Description

This mechanism uses an encrypted channel to protect the content in addition to the features of RTMP.



3.3.2 Implementation requirements

Implementations are not required to support, and content providers should not attempt to deliver, content at bitrates exceeding 4 Mbps using this method. It is expected that on many devices, software-only implementations will be able to meet this requirement.

This content delivery mechanism does not provide means to change the state of output control mechanisms from the default specified in section 2.8.

4 Presentation engine specific content protection: MHEG-5

Devices shall support the *ICEncryptedStreamExtension* defined in D-Book 6.2.1.

5 Presentation engine specific content protection: W3C

No presentation engine specific content protection mechanisms are defined for W3C.