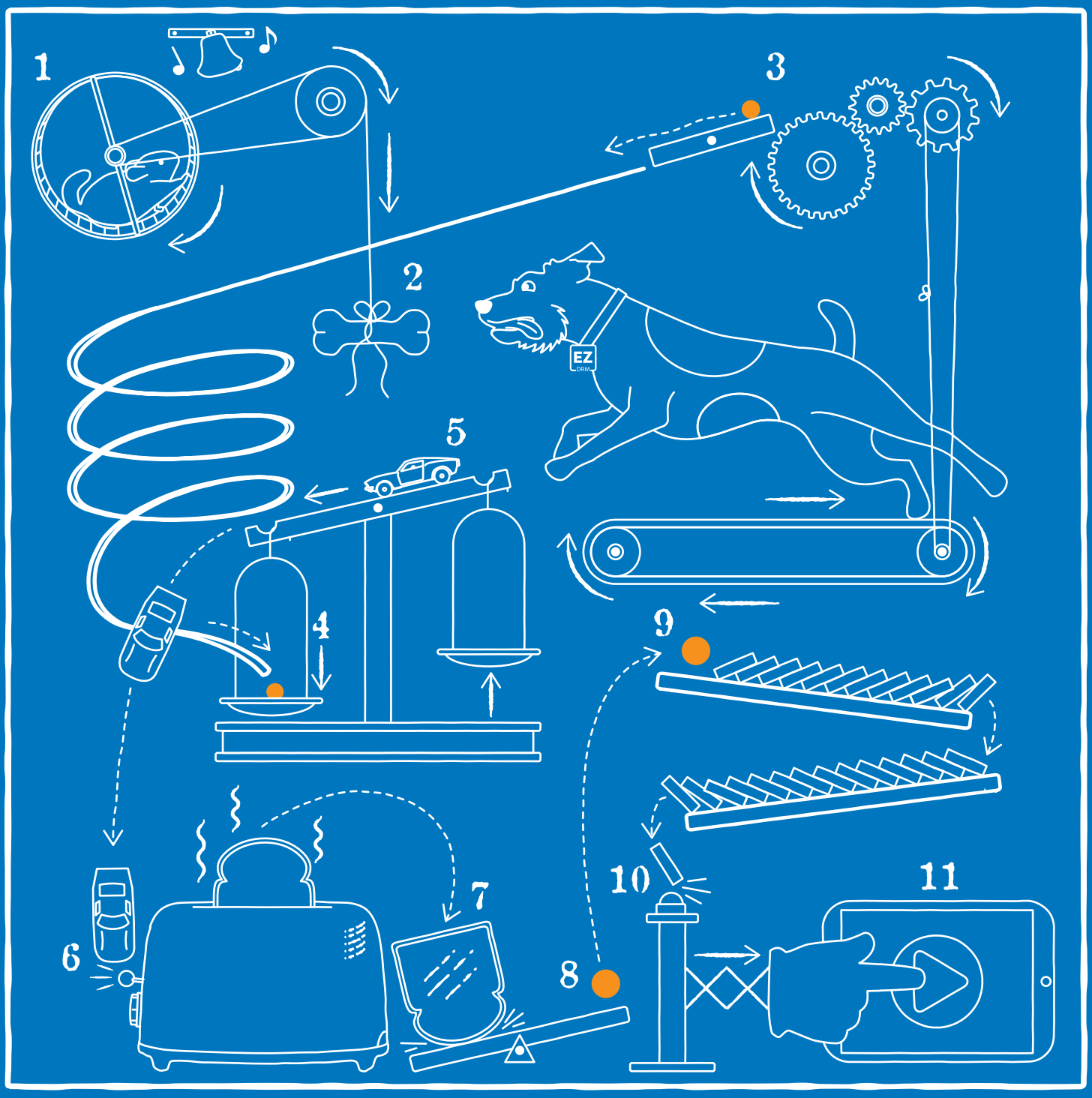


DRM as a Service: What Makes it Tick?

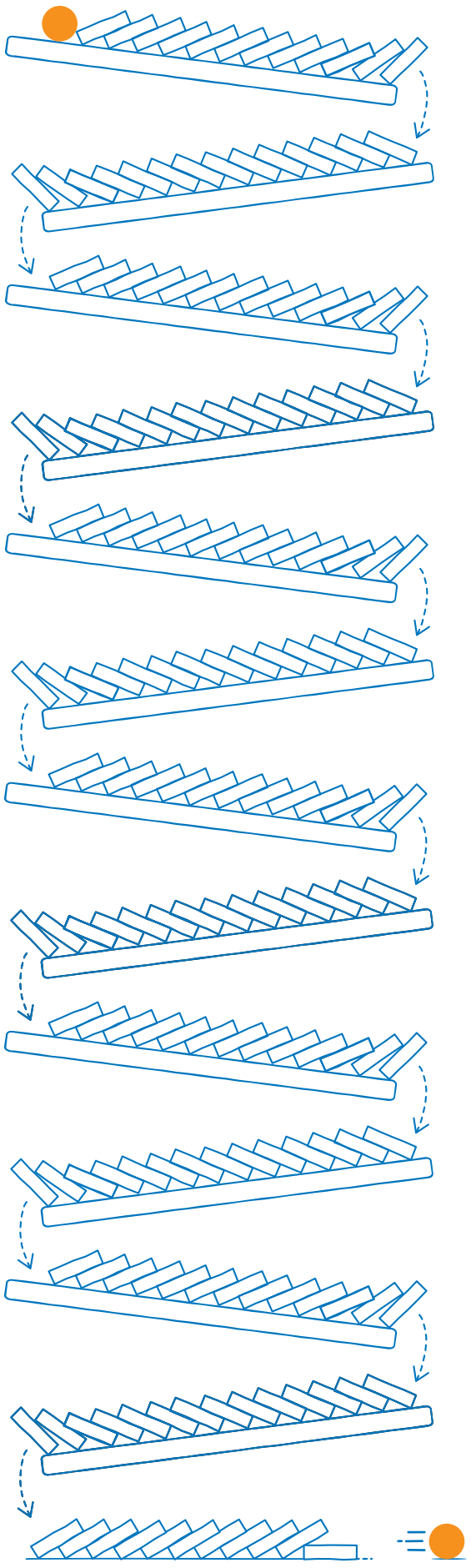
What's *Really* Behind the Award Winning EZDRM Service

EZDRM®

Digital Rights Managment.
Simplified.

At a time of great upheaval and opportunity in our industry, we at EZDRM believe that we have found a good moment to publish a comprehensive version otf our FAQ / Question and Answer information. Over the years of our DRMaaS business, the mission has been to help cut through the apparent complexity and jargon that surrounds the topic of security. Just like the EZDRM services themselves, we have found that helping to simplify the topic of how Digital Rights Management (DRM) has evolved also helps to communicate the benefits. The result is an evolving document that addresses a wide range of recurring topics related to the technical and business issues of DRM, and more importantly, how it is used in today's video services marketplace.

We hope you find the information we have gathered here to be comprehensive and useful, but if you have additional questions on this general topic please don't hesitate to send us a note (by using a convenient form on our site or by sending an email to simplify@ezdrm.com) and we will be happy to add some helpful comments.



Introduction to Digital Rights Management

Security as a whole is pretty important to any thriving video service business. There are large sums of money invested in the content, infrastructure and marketing (and so on), as well as a significant amount of personal subscriber data, that must be protected. Digital Rights Management (DRM) is actually an essential component of the video business, although most people think of it purely as a technology. Any video service business has to manage the granting of rights to view video – and those rights are only granted for short periods of time. For example, in a subscription video service, if the subscription dues are paid for March, but not in April, then the viewing rights should cease, when that new month starts. If a video is rented in a pay-per-view transaction for a period of 3 days, then viewing outside of this 72-hour window should not be allowed. Managing rights can therefore be tied very directly to payment, as a part of the that service business model. The technology of DRM ensures that these viewing rights can be securely granted and that the rules can be enforced by the video operator. Since even large digital files can easily be stored and copied these days, the video streams or downloaded files should not directly readable or editable. This also protects the operator of an advertising supported service. If the video service content can be stripped of its advertising load, then the business for that service falls apart. If clear copies of valuable content are circulated without limit, then the whole content value chain – from performer to service operator - is undermined.

How does a service operator protect their own interests with DRM?

Firstly, ensure that the video content is encrypted in a secure fashion on the service side and remains in that secure, encrypted form during the process of delivery. The keys used for this encryption must be specifically protected – this is the server component of the DRM scheme.

Secondly, only deliver video to devices that are trusted. Trust here, comes from knowledge of the consumer (through, for example a login or other credentials) and consideration of the consumption device. In general, devices that can be trusted for playback are those that incorporate a verified implementation of one or more DRM clients.

Thirdly, only deliver the information necessary for video playback to these trusted consumer devices in a highly secure form – one that does not permit interception or tampering – and that explicitly controls the time window and other constraints on the playback process. This secure packaging is the core nature of a DRM license.

Fourthly, only permit the process of video playback in conjunction with a current valid DRM license to eliminate any possible illicit capture of unprotected video during consumption. A fully secure integration of the video player, the DRM client and the device video hardware is critical here.

It's clear from this description that there are client-side and server-side components to any DRM solution that should be described in more detail. In the body of this FAQ document, we will attempt to offer more explanation and context. But we could say that, from a service operators' perspective, today's DRMaaS approach largely conceals a lot of the gnarly details of the technology behind the curtain of a cloud API.

What part of the DRM process happens on the server or cloud side of the video delivery service?

The short answer to this question is that the server-side components of DRM are concerned with encryption of the video content in a way that makes it impossible to view without also having access to an appropriate license. The encryption or packaging process actually is a part of the video delivery workflow and takes place between video encoding and actual video stream delivery. The key values used for encryption are generated together with a video asset identifier and supplied via a secure API to that encryption process. The important point about this arms-length separation is that the encrypted video and the database of keys used to protect it are never stored in conjunction with one another. In fact, the encryption key/video asset identifier pairs are only stored in the context of the DRM server-side component, which is regarded as specifically secure for this purpose, whereas the now protected video files can now be made available via standard, highly optimized web request/response protocols.

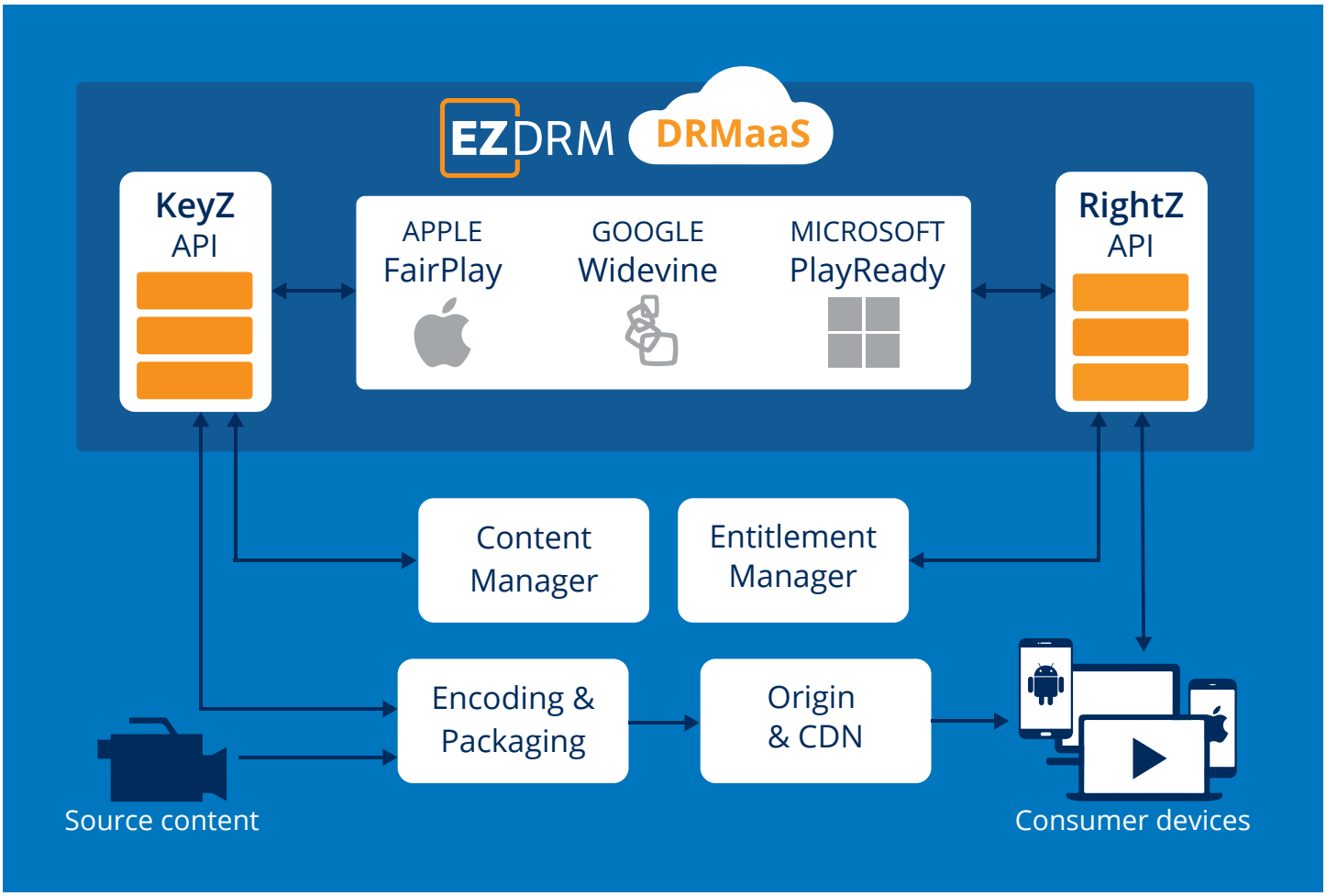
What part of the DRM process happens on the client/device side in an app or browser session?

The goal of every video service provider is responsive, smooth video playback on the widest possible range of

client devices. The client-side components of the DRM process support this goal, while acting to protect and/or enforce the business model behind the video service. The process begins, when a consumer uses a device app or a video web session to select a piece of content that they want to play. The request to access that set of video files kicks off a number of parallel processes. The first, most obvious process is that of streaming the video content itself. Because these files are relatively large, the initial download requires a little startup time. During this startup phase, the video player recognizes that the content is DRM protected and requests the device DRM implementation to obtain a required playback license, passing along the video asset identifier information, details of the device itself and any other required parameters. The DRM client then makes a secure request to the DRM server component to receive the appropriate license. Before responding to this request, the DRM server checks the rights that this consumer has for video content playback and, if satisfied, creates a DRM license object that's specific to this content, this request and the device that is requesting playback and delivers it to the consumer device. The client video player uses the license to play back the content for the end user.

What exactly is a DRM license?

In the body of the FAQs below, we try and explain the general technical nature of a DRM license, although the actual format is a highly proprietary secret in each of the major DRM systems in use. At the highest level, the process described above separates out the distinct concerns of delivering the video content itself from the communication and rights checks that simultaneously delivers the necessary decryption keys and the rules about playback constraints. The large and continuous stream of video content is delivered from one cloud resource, while the license that permits playback is requested and delivered by a separate, almost instant one-time transaction with the secure DRM service. To deliver service efficiency, the video content streamed to all devices is essentially identical, while the license required to view that content is created in real-time to be very specific to each playback request and the device that made it. [See later FAQ section on license technology].



Frequently Asked Questions on DRM and Security

Q: What is DRM as a Service (DRMaaS)?

When you are setting up a video service, there are many components you need to put into place. In past years, the deployment model was to invest in a specialist data center and racks of equipment to do video acquisition, video encoding, subscriber management, security and so on. The predominant model in today's market is to use an integrated suite of cloud-based components to provide the same functions - but without the noisy fans of the server machines!

Since Digital Rights Management (DRM) is a specialist area of expertise and needs specific attention to licensing and security amongst other things, usage of a cloud-based service that supports the DRM function you need is a very convenient simplification of the overall deployment setup. The benefits of using a service for DRM implementation include:

- No capital costs - payment primarily on a transactional, periodic basis that scales with the use of the service
- Fast time to market - the infrastructure is already set up and only requires integration logic to tie to the rest of the service workflow
- Easy integration - well documented and fully tested templates
- Easy scaling - the management of the service includes the addition of resources as required load grows
- Access to specialist expertise when required - you always have a resource available to keep things running at peak performance

Q: Why is DRM useful to distribute video content?

Anyone in the business of commercial video distribution, be it in the enterprise or entertainment space, should be fully aware of the imperative to protect such video from unauthorized viewing, copying or re-distribution. If you don't have such protection in place, its increasingly hard to create a viable business around the video service.

Content producers (including movie and TV industries) are especially protective of their video assets and, if you try and license such assets for your service, they impose strict rules on distribution to ensure that appropriate protections are in place. DRM solutions are the core of such protection mechanisms. They are used to encrypt video before distribution AND control the distribution of the vital license information that limits viewership to a specific time window and to trusted devices and consumers.

The specifics of DRM requirements and constraints can vary with the quality (or value) of the assets being licensed. These requirements can dictate what content can be delivered to particular classes of devices, such as PCs, smart phones and tablets for example. In general, and with today's technology landscape, SD (480p) playback can utilize software-based DRM while HD (720p+) playback typically requires a hardware enhanced DRM implementation (see FAQ section on hardware security). Providing 4K/UHD quality content also typically requires hardware-secured DRM along with additional security requirements beyond DRM, including watermarking (See FAQ section on watermarking).

Q: How is DRMaaS managed?

DRM as a Service is a fully managed software solution, which means that you, as a video service provider, do not need to be concerned about:

- System setup - all initial service configuration and reporting management is undertaken as you set up your account
- Security - all system data is maintained in a fully hardened environment, so there is no concern over unauthorized access to key data or business critical transactions
- Privacy - Exposure to regulatory issues around personal identifiable data is limited by design in order to ensure compliance
- Monitoring - the management of service operation is covered by expert engineers on a 24/7/365 basis. Any issues that arise with operation or performance are flagged and resolved before becoming business critical.
- Resilience - using state of the art cloud resource management techniques, resiliency and redundancy is automatically addressed. Issues that arise with communications or compute resources are resolved in real time.

- Upgrades - an important part of this business is keeping up to date with the evolving technology. As the DRM versions from the various vendors evolve and new features are released, the integrated service can be ready on day one.

DRMaaS is accessed by an easily implemented API from the encoder or packager solution in use. The majority of commercial encoder/packager solutions are already supported by EZDRM.

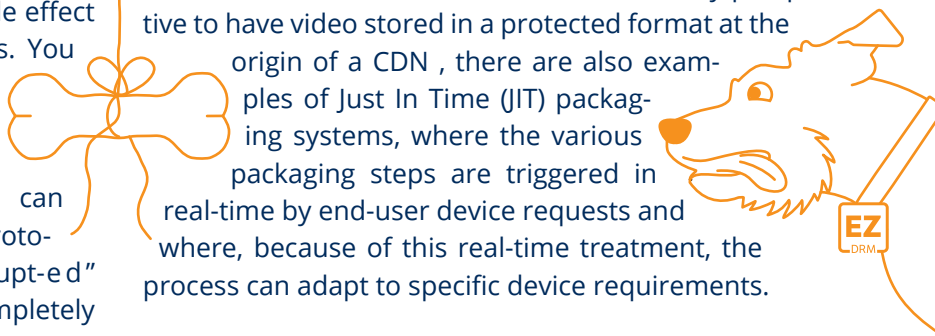
Q: What does the video encryption process protect?

Compressed video and audio playback are universally supported on today's computers and consumer hardware. But the process of compression makes the data in the video file or stream exceptionally sensitive to alterations in its binary content – only minor corruption of the video data is needed to have a massive ripple effect on the decompression and rendering functions. You can see that sometimes, when a video you are watching glitches and dissolves into a mass of seemingly randomly colored pixels. Such data changes, that disrupt the playback function, can happen accidentally - when communication protocols break for example. Or the video can be "corrupt-e d" deliberately at its source by partially or completely encrypting the content using a scrambling key. Unlike random corruption of files or streams, this encryption is a very precise process using controlled scrambling keys and a well-defined algorithm. The whole design concept of this encryption process is that it is, of necessity, 100% reversible. With this premise, under the right circumstances, it is possible to reconstruct the original high-quality source material from the scrambled stream or file.

Scrambled/encrypted video content is essentially valueless without knowledge of how to reverse the encryption process. It can't be played or redistributed. Effectively, scrambled video content often uses a different scrambling key for each time period of a live stream, and certainly for every individual video asset in a library. Reversing the encryption process, of course, requires access to the scrambling key or keys used for protection and knowledge of the algorithm used. Controlling this access is the essence of video security and why it is core to the video service business.

Q: Where does video encryption happen?

In most streaming video workflows, the option to encrypt video is a processing step between the output of the video encoder and the input of the distribution platform. The operations that are often combined are segmentation of video streaming into delivery packets, running encryption over those packets and building of a manifest file that references both the packets and the identifiers necessary for video players to request a playback license. The overall operation is often termed "packaging". When DRM is used, the actual key values are requested from the DRM service (which assures appropriate randomization etc.) and are paired with the content identifiers to be used to retrieve those keys at playback. Wherever the keys are generated, the storage of those keys is wholly trusted to the DRM service.



While it makes the most sense from a security perspective to have video stored in a protected format at the origin of a CDN , there are also examples of Just In Time (JIT) packaging systems, where the various packaging steps are triggered in real-time by end-user device requests and where, because of this real-time treatment, the process can adapt to specific device requirements.

Q: What about content watermarking?

When video content is watermarked, typically during encoding (but also potentially at time of individual device delivery or even during specific device playback), it means that some form of individualized identifier information (often termed a payload) is added to the main video content. The identifier payload is - by design - intended to be subliminal or imperceptible, so that it does not interfere with the viewing experience. But because the payload and the video are combined on the screen, any capture or copy made will also include that payload. When an illegal copy of a video is re-distributed, the payload can be extracted and acts as a way to trace the source of that copy to provide an additional form of security mechanism that extends beyond the envelope of encrypted delivery.

For more general information on video watermarking please refer to the Digital Watermarking Alliance.

Q: What is ABR?

Adaptive Bitrate (ABR) streaming has been a core enabler of the streaming video revolution by offering a

way to deliver a good user video consumption experience across a wide variety of networks and device types. ABR created a way to solve several practical problems with first generation video delivery systems and the IP protocols they used, including traversal of in-home broadband routers, security and inconsistent user experiences.

From a technical point of view, ABR is a method of video streaming that uses HTTP as a core protocol and where the source content is encoded at multiple bit rates. Each bit rate stream is then sliced into small multi-second segments. The streaming client is informed of the various available bit rates using a coordinating file termed a "manifest". A client device initially reads the manifest and uses that information to request initial segments of video. During the request process, the time taken to download each segment is measured and compared to the playback time that the segment requires. If the player client finds the download speed permits use of a higher quality, higher bitrate stream, it "gears up" to request those segments. Conversely, if the download time is longer than it would take to play a high-quality segment, the client "gears down" to request smaller, lower bitrate segments. Segments themselves are encoded in such a way that the "joins" become invisible to the viewer, so the client can over time optimize the download rate and viewer quality and adjust to compensate for any short-term variations in bandwidth. Since the optimization is continuously in-progress and is undertaken from the client devices point of view, the technique is very effective at providing a viewer experience that's free from stalls and glitches.

Although originally commercialized by Move networks, ABR came to prominence – or even dominance – through implementations by Microsoft (Smooth Streaming), Apple (HTTP Live Streaming) and in the MPEG-DASH standard.

Q: What is HLS?

HTTP Live Streaming (HLS) is an ABR streaming communications protocol implemented by Apple, that become core to the video system on iPhone and iPad devices. HLS naturally supports both live and on-demand content streams, where the segments of video at the various bitrates were historically created in "chunks" of small MPEG2-TS files. The manifest file is a text file with an .m3u8 extension (based on prior MP3 streaming formats). Apple standardized the format through draft IETF RFC documents and these documents have gone through many revisions. As a practical matter, the HLS format has been wildly popular for streaming video services and still dominates the delivery world.

Q: What is MPEG-DASH?

In April 2012, ISO (the international standards body, which had already given us the core media foundations of MPEG-2, MP3 and HEVC etc.), finally ratified the version of an adaptive streaming standard: MPEG-DASH (DASH stands for Dynamic Adaptive Streaming using HTTP). The participating companies in the MPEG-DASH standardization (including Microsoft, Apple, Netflix, Qualcomm, Ericsson, Samsung, and many others) saw a vision of interoperability and convergence required for large-scale market growth in streaming video that trumped the prior emphasis on proprietary and competing streaming formats based around HTTP. In terms of sophistication and flexibility, DASH leapfrogged the existing multiple proprietary solutions with a single industry-defined open standard.

Of course, publishing a standard only marks the beginning of the work to achieve detailed interoperable usage. The baton for a lot of the practical effort was passed to an industry forum to promote and catalyze the adoption of MPEG-DASH. The DASH Industry Forum (DASH-IF) is filled with member companies, including EZDRM, who are realists about the DASH deployment challenges. They are willing to take on the work of creating recommendations, filing bugs, and attending plug-fests and interop events with the belief that their business and the Internet streaming market at large will benefit a great deal from a convergence around DASH. For further information see https://dashif.org/.

Q: What is CMAF?

The Common Media Application Format (CMAF; officially MPEG-A Part 19 or ISO/IEC 23000-19) is a newer initiative, that builds on MPEG-DASH standardization, but also attempts to reconcile the de-facto adaptive streaming standard (HLS as defined, implemented and evolved by Apple) with some of the DASH-related advances and move the ball forward to a truly secure common file format that can be used as the origin of all streaming delivery.

Q: What is Common Encryption (CENC)?

In an effort to reduce fragmentation of the video delivery equipment and software market, a standardized algorithm for streaming video file encryption has been creat-

ed. It is called Common Encryption (also referred to as CENC). CENC is an ISO 23001-7 standard that defines how encryption will be applied to different sections of a video file format and what scrambling algorithm will be used with the encryption key(s). Ultimately, this CENC format enables the same secure content to be distributed to numerous playback devices/platforms, all of which can use the defined algorithm to restore the content to playable form. What isn't standardized in CENC is how the scrambling keys themselves will be delivered to the playback device - this is left to the technology of distinct DRM clients in each device. Individual DRM clients retain responsibility for the security of aspects such as license distribution, rights mapping, and compliance, which means these processes can be implemented in a proprietary fashion in individual DRM systems. (See the FAQ section on license delivery.)

Q: Browsers or Apps?

How a given service chooses to present their video on any given device, and the accompanying elements of the service's user interface, is a very important question, but one that sometimes has political, as well as technical overtones. In general, downloadable apps are used for commercial video services on mobile devices, tablets and TVs, while desktop/laptop computers often use a browser environment. There is a lot of debate over the merits of the two approaches in terms of user experience, considering startup time, service latency and device efficiency for example. The standardized video format of MPEG-DASH was originally defined around the need for in browser consumption, although the usage profile is very much a moving target at the present time.

Q: Which browsers support DRM for video services?

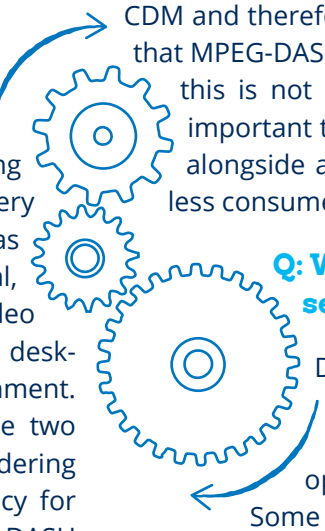
All modern browsers from different vendors on each consumer platform have adopted the HTML5 principle of including support for video as part of their native code and then extending that support to provide mechanics for playback of DRM protected video (see FAQ section on EME). Refer to the table on our DRM comparison page.

Q: What are Encrypted Media Extensions (EME)?

Encrypted Media Extensions are a set of JavaScript API specifications that provide a way for the code of web pages to interact with DRM license servers and the built-in security mechanisms of browsers (see FAQ section on CDMs) to play back DRM protected videos within a web page. A long-standing political tussle has delayed formal standardization of these APIs by the W3C, but de-facto support exists in all commercial browsers. See the EME specification.

Q: What is a Content Decryption Module (CDM)?

A Content Decryption Module is an implementation of proprietary DRM client code inside a browser that lies behind the standardized Encrypted Media Extensions (EME) implementation. As originally conceived, any browser could support multiple CDM implementations, but politics and commercial rivalry has limited each of today's commercial offerings to support of one single CDM and therefore a single DRM format. Given the way that MPEG-DASH streams and the use of CENC interact, this is not as limiting as it may sound - but it is important to operate an MPEG-DASH video service alongside a multi-DRM service to provide a seamless consumer playback experience.



Q: What about DRM hardware security?

DRM functionality in devices can be supported wholly via software, for example, by being built into a device's operating system or a library module. Some devices have DRM clients that have been integrated in a way that uses security features of the underlying chipset to pipeline the video playback system. This provides a 'hardware protection' layer to the logic of the DRM client and protects any decryption keys or unencrypted video from being intercepted by malicious code running in parallel with the DRM client software. The hardware support is offered by devices that use ARM TrustZone or similar secure processor element in their architecture. The details of how security is implemented at this level are beyond the scope of this FAQ, but fundamentally tie the DRM logic to the hardware implemented identifier of the device chipset and the individualized secret keys programmed alongside this identifier at manufacture time. Sometimes the shorthand way to refer to this setup is a "hardware root of trust".

Q: What is CPIX?

CPIX stands for Content Protection Information

Exchange, a rather bland term for a standard that brings very exciting changes for the media industry. Driven by the DASH Industry Forum, CPIX is designed to create operational efficiencies and reduce the launch time for your OTT services.

CPIX is used as a base mechanism to exchange key values between the DRM service key manager and encryptors or packagers. Historically, every service vendor used its own proprietary interface and DRM-specific APIs to handle this information exchange. Hence, switching from one solution to another could be complex and error prone. CPIX has become a common denominator in such interfaces. In that sense CPIX breaks the vendor lock-in for operators.

A feature of CPIX that is becoming increasingly useful for premium services is the ability to specify a multi-key encryption process. CPIX multi-key serves to protect stream content at different resolutions with different key values so that, for example, a variety of business models can be supported with a single stream asset. See more details on our website.

Unlike some prior efforts, which tended to try and focus on a single streaming ecosystem, CPIX addresses both DASH and HLS streaming protocols. This enables operators who adopt CPIX to securely stream video to virtually every type of connected device.

Q: Is SPEKE the same thing as CPIX?

Building on the standardization effort with SPEKE, AWS introduced the Secure Packager and Encoder Key Exchange, or SPEKE for short. SPEKE is an open API specification that fully streamlines the way Digital Rights Management (DRM) systems integrate with packers and encryptors (in the SPEKE context "encryptors" encompass encoders, transcoders, and origin servers that interact with DRM systems to obtain and use encryption keys).

SPEKE delivers a standardized API for key exchange between encryptors and DRM systems for both live and on-demand media workflows, while allowing media customers to use any SPEKE-enabled key server or encryptor in on-premises, cloud, or hybrid infrastructures. SPEKE incorporates the CPIX specification and builds this foundation to address practical issues, such as service authentication. As such, the API eliminates the need for complex integrations between proprietary multi-DRM APIs and encryptors from different vendors.

SPEKE-enabled DRM services should work with any SPEKE-enabled encryptors out of the box. SPEKE supports all major DRM technologies (Apple FairPlay, Microsoft PlayReady, Google Widevine) as well as Clear Key for HLS, MSS, and DASH formats. More than a dozen vendors have already endorsed or demonstrated the use of SPEKE API.

Q: What is PlayReady?

PlayReady, by Microsoft, is probably the most influential and broadly supported DRM solution in today's marketplace. It was introduced in 2008 as a more device portable and standards-centric evolution of the prior Windows Media DRM system. PlayReadyhas been adapted and extended in various ways since, that point to support streaming vs file download, live vs on-demand, and domain vs personal licenses.

At least partly because of its open approach to device licensing, PlayReady has a wide and deep ecosystem that includes implementations on a broad array of devices and operating systems to protect premium content streaming while driving the optimal user experience.

Devices and environments that embed PlayReady DRM clients include:

- The Edge and IE browsers
- The Windows OS
- Xbox
- Most smart TVs
- Roku and Fire TV streaming devices

The license server components of PlayReady have also been widely implemented as part of commercial DRM systems, including EZDRM's Universal DRM.

EZDRM supports both Microsoft PlayReady and Windows Media Rights Manager technologies. Both Microsoft Smooth Streaming (PIFF) and MPEG-DASH (DASH) are supported content delivery methods.

Q: What is Widevine?

Widevine DRM was originally developed and marketed by Widevine Technologies. The company was acquired by Google in 2010 and Widevine DRM has been vertically integrated across the major Google offerings since that point. Currently, Widevine is the default DRM technology for:

- Android tablets and phones from all vendors
- Android TV devices
- YouTube
- Chrome, Firefox, Yandex and Opera browsers
- Chromecast dongle devices

There have been two versions of Widevine DRM deployed – the so called modular and classic versions. For all practical purposes, Widevine Modular DRM is the major branch of the technology and supports the standards-based world of DASH, HLS and CMAF streaming services. Widevine Modular also supports the latest client device security HW features and functions to offer Level 1 grade security.

The EZDRM Widevine DRM service enables license generation, secure distribution and protected playback of content on any Android and many other consumer devices to ensure revenue generating services keep streaming to the customer's desired device.

Q: What is FairPlay?

Apple's FairPlay® Streaming (FPS) DRM provides secure video playback through the full wide range of Apple devices and operating environments. The client software is not openly licensed, but is the default security environment across at least:

- iOS phones and tablets
- Apple TV
- The Safari browser
- MacOS

On these devices FPS supports HLS streaming, plus some support for MPEG-DASH and CMAF formats.

The EZDRM hosted FairPlay Streaming service enables users to enjoy the best video entertainment experience with a clear route to obtain rights to premium entertainment from Hollywood Studios and broadcasters and protect your revenues preventing unauthorized use and piracy.

Q: What is Clear Key?

Clear Key (or sometimes Clearkey) media encryption has been described by some in the industry as the poor man's alternative to DRM. Nevertheless, the support of a Clear Key mechanism is a highly appropriate component of every DRM license management service.

Clear Key represents is foundational layer of a streaming protocol that offers the full mechanics of media encryption and decryption for protecting audio/video content in transit, but avoids complex and proprietary delivery protections for the encryption key itself. The typical support approach is for the encryption key to be made available directly in the manifest file of the ABR stream or via a simple http/https call to a key server identified in the manifest file.

This Clear Key approach was very broadly supported in the initial implementations of Apple's HLS streaming protocol [RFC link] and became very popular because of its simplicity, although it offers very little in the way of the content security that is typically required for sophisticated commercial services as we see them today. In the context of the basic HLS protocol it is often referred to as Apple AES-128 encryption as a shorthand.

MPEG DASH adopted the mechanics of Clear Key as vendor-neutral lowest common denominator approach to enabling the Encrypted Media Extensions (EME) that protect playback of video streaming within the latest generation of browsers.

Clear Key handling is required within EME support for all browsers, but it could be seen primarily as a reference mechanism, rather than a commercial solution. Using DASH Clear Key, media can be encrypted with a fixed key at the server, and the key value can be signaled in the Media Presentation Description (MPD) manifest file. The practical effect is to enable encryption on the server side, and decryption in the player code without the complexity of obtaining and managing a typical DRM license object.

Q: What is WisePlay?

WisePlay DRM is a relatively new branch on the tree of DRM technologies available to audio and video services and is being developed and marketed by Huawei Technologies.

Momentum has grown rapidly around this offering recently, primarily driven by the current US sanctions policy deployed against Huawei. The implication of these sanctions is that none of Huawei's products can include US sourced technology – and especially not US sourced cryptographic technology such as that included in the major DRM implementations from Widevine, Microsoft

and Apple. Without being able to offer DRM protection in the client devices, all major video entertainment services would effectively be unusable on Huawei phone and tablet devices - which at present represents nearly 20% of the world market (globally number one ahead of Samsung and Apple).

WisePlay is a Huawei home-grown client DRM technology that is built into those millions of Huawei devices and that offers an alternative to the US DRM sources. Significantly, this is not a wholly proprietary concept, but inherits the credibility of the broader and long-running ChinaDRM standards initiative.

Operationally, WisePlay adopts many of the features and functions of the other major DRM vendors. Media content can be packaged and streamed in DASH-fMP4, HLS-fMP4, and HLS-TS formats with standard encryption schemes that include CENC, AES128-CTR, and AES128-CBC. Because of this standards foundation, particularly support for Common Encryption (CENC), a single stream can be protected in such a way that player devices can use their own embedded DRM scheme to obtain DRM licenses. With multi-DRM packaged content Huawei devices can play back such streams using their WisePlay DRM client just as easily as Samsung devices can use Widevine to see the same content. Additionally, as you might expect from a high-tech giant like Huawei, WisePlay uses hardware-based security mechanisms in the device DRM code to provide security levels that meet or exceed the requirements set by major content licensees like the Hollywood studios.

When built within a multi-DRM service such as EZDRM Universal DRM, the use of the WisePlay technology is almost completely transparent and enables video service delivery to Huawei devices in parallel with devices of other mainstream vendors.

Q: What is a DRM license?

The security of the encryption keys used to protect video content would be easily compromised, if the communication between the DRM client on a consumer device and the secure DRM key service was not itself heavily protected. This protection required goes way beyond the typical communication security of a web protocol such as SSL. To address the specific needs of DRM, the decryption keys are delivered to client devices in the form of a communications object known as a DRM license, that incorporates key management that is specific to the video service and, at the highest level, to the hardware of the playback device itself.

The overall goal is to ensure that the video encryption cannot be compromised by interception of the license communication or the process of unwrapping that license to actually decrypt the video content during playback. It is also convenient to include, wrapped up in this same protected license object, all of the information about restrictions on use of the license keys - such as the time period over which the license is valid, the constraints on protecting video outputs and so on.

In fact, the concept of a DRM license in the form of a file is pretty anachronistic and dates from the early days of Internet digital audio and video distribution where the objective of a DRM system was to protect downloaded files in an almost wholly offline consumption model. With today's near continuous online streaming model, it might be a lot more straightforward to check user rights in real time during any section of the playback of content. This historical precedent leads to a number of undesirable effects in media system, such as the inability to rescind the issue of a license. It does, however, provide a way to distinguish between transient and persistent rights for content consumption.

Q: What's the difference between a transient and a persistent DRM license?

Generally, non-persistent licenses are used for immediate playback of content and can also be thought of as "in-memory" licenses. Non-persistent licenses are intended to last for only as long as the current consumption session, or a single playback.

Persistent licenses can be stored in non-volatile memory on a client device after they are received and are used for any playback session until time-based rights restriction embedded in the license is reached. For example, a persistent license can be used to play back downloaded content while the device is offline. Enforcing protection mechanisms in these scenarios require a local trusted reference for time.

Both persistent and non-persistent licenses include a range of rights and restrictions set by the content licensor.

Q: What is delivered during a DRM license request?

In a commercial video service, the video content itself is scrambled using one or more encryption keys (see FAQ

section on video encryption), which are stored separately from the encrypted video within the DRM service database. However, the encrypted video content must contain a cleartext identifier of the DRM system or systems, where the key can be obtained (typically a URL), together with an identifier (often numerical), that can be used to select the right key from the database maintained by that DRM system.

Without going too far into the details of video player logic, when a video player (web browser or app) attempts to render an encrypted video stream, the video player process recognizes the need for a license and requests it from the remote DRM service. This request can also contain client details, that can be used by the video service provider to authenticate the user and determine whether or not this client has the rights to play this content at this time. If the license request succeeds, the video plays.

Q: How does DRMaaS pricing work?

The pricing approach of DRM services is designed to be straightforward and transparent. It also helps that the costs of security are intended to scale very gradually with the size of the underlying video service business.

The DRMaaS system keeps track of the number of DRM licenses issued for content associated with a specific video service. In general, a DRM license is required, and is delivered, for each play of each piece of content on each individual consumption device. So, if "Game of Thrones" is watched on an iPad, a TV and a PC on a given day – this will require a distinct DRM license for each device. If content is viewed again the next day, a new DRM license will be required for each device.

The pricing of individual DRM licenses scales according to the volume required per month.

We hope you find the information we have collected here useful, but if you have additional questions to add to this general topic please don't hesitate to send us a note using the form on our web site and we will add further answers. (And give you the credit!)

Thanks for reading,

The Team at EZDRM