



Smartlabs

SmartMEDIA
Content Delivery System

Functionality Overview

Contents

1. Introduction	3
2. VoD Services	6
3. nDVR Services	9
4. Transcoding	14
5. Multicast Content Protection	18
6. Load Balancing and Failover	21

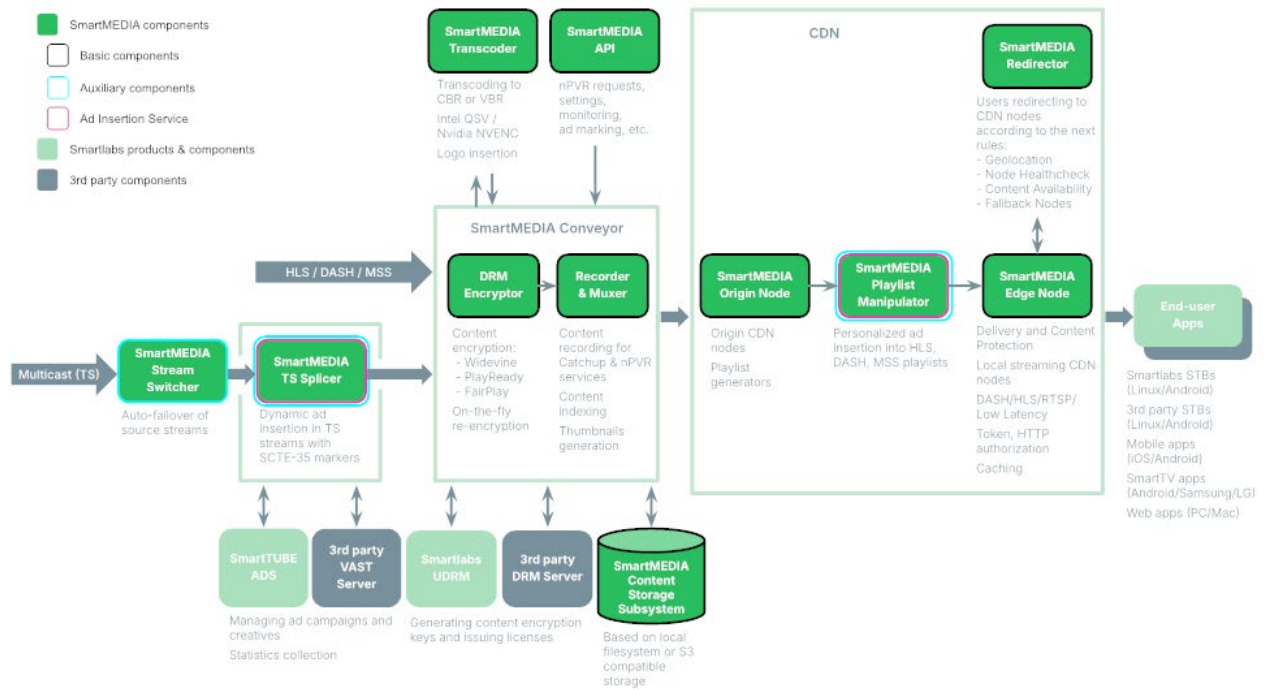
1. Introduction

Review of SmartMEDIA

The SmartMEDIA media server, developed by SmartLabs, is designed to deliver video content over IP networks. It provides transcoding, segmentation, encryption, recording and delivery of audio/video content for both managed IP (IPTV) and unmanaged IP (OTT) networks.

SmartMEDIA supports a variety of streaming protocols (MPEG-DASH, HLS, RTSP), video resolutions (from SD to UHD) and codecs (H.262/MPEG-2, H.264/AVC, H.265/HEVC, various audio codecs, etc.), allows you to deliver media content to a wide range of subscriber devices and players, including set-top boxes, browsers, Smart TVs, Apple iOS devices, Android devices, etc.

Solutions based on SmartMEDIA can automatically distribute the load between video servers depending on the client's location and requested content for high performance and fault tolerance. This allows you to build distributed video content delivery networks (CDN) in particular.



SmartMEDIA architecture (click to enlarge)

Key Features of SmartMEDIA

- High scalability with support for both centralized and distributed solutions.
- High reliability due to the ability to build failover clusters with no single point of failure.
- Deploy the hardware platform quickly and easily with standard servers and storage systems.
- A wide range of supported formats (from SD to UHD), codecs (H.262/MPEG-2, H.264/AVC, H.265/HEVC and others), as well as integration with popular content protection systems and subscriber devices.
- The SmartMEDIA media server software is running Linux on top of the x86_64 platform.

Software Versions

The difference between the **SmartMEDIA Legacy** and **SmartMEDIA Ultra** versions is the support for the feature "*CENC->CVBS On-the-fly Re-encryption*" by the Ultra version.

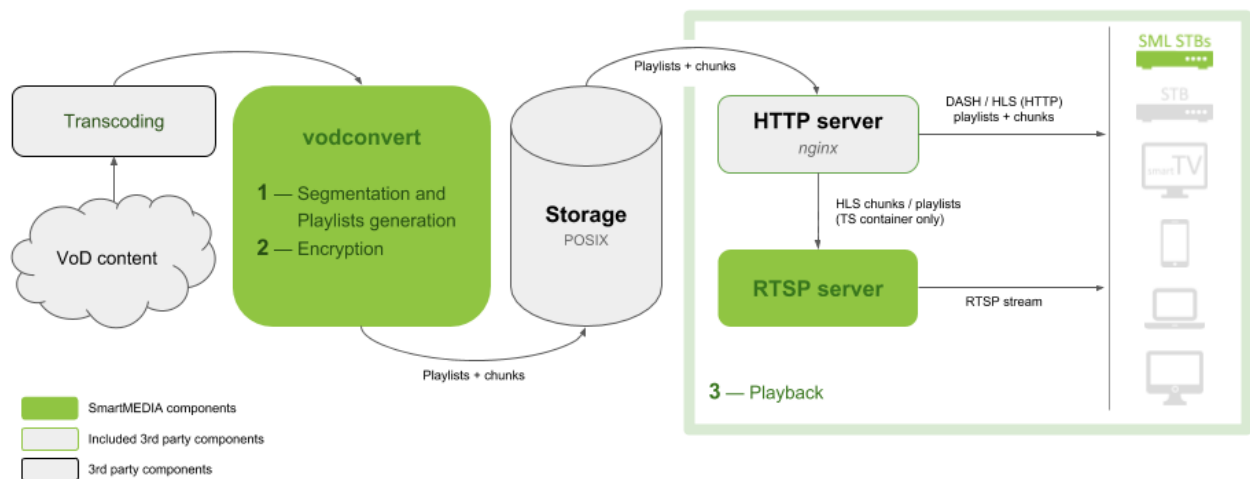
This feature is intended to re-encrypt MP4 chunks prepared for adaptive streaming (HLS or DASH) from one encryption scheme to another "on the fly", i.e. without saving the re-encrypted copy to the repository. This saves disk space by storing only one copy of encrypted content (for example, for Widevine DRM). At the same time, players using a different content protection system (for example, Apple FairPlay DRM) can also decrypt and play this content. Currently, reencryption from the cenc scheme (AES-CTR) to the cbcs scheme (AES-CBC with template encryption) is supported according to the standard ISO/IEC 23001-7 "Common encryption in ISO BMFF files".

2. VoD Services

Using the SmartMEDIA media server, IPTV/OTT service providers can develop a wide range of services for their subscribers, the entire list of which can be generalized to the following services: LiveTV (multicast), VoD and nDVR.

VoD Services

VoD (Video on Demand) is a set of services for targeted video content delivery (for example, movies, serials, etc.) to end-user devices using IP networks. For VoD-services the content should be properly prepared (transcoded, segmented, encrypted and indexed) and stored in a storage for future transfer to the devices. These steps are implemented in SmartMEDIA as follows:



VoD Content Preparation Stages

1. Segmenting and Playlist Generation

VoD content should be segmented into chunks according to MPEG-DASH and HLS protocols. Chunked content will be packaged in TS and/or MP4 containers and recorded to the storage.

Requirements for Source Files

Supported containers:

- SPTS MPEG2-TS (ISO/IEC 13818-1, ITU-T);
- MP4/ISO BMFF (ISO/IEC 14496-12 – MPEG-4 Part 12);
- MKV (Matroska).

Supported codecs:

- AAC, AC3, DTS – for audio;
- H.262/MPEG2, H.264/AVC, H.265/HEVC – for video.

Additional requirements:

- If you're going to use adaptive bitrate streaming (ABR), all video streams should have the same GOP structure and have synchronous key frames (I-frames).
- The value of DTS counters for audio and video samples should not differ by more than 1.5 seconds.
- The value of DTS counters should increase monotonically.
- If the PMT table in the MPEG2-TS container was changed, a table version change is required.
- There should be no [CC](#) errors in the TS-stream.

POSIX-compatible file systems are supported.

Playlists with chunk references and other metadata will be generated as well as media files. Playlists will be stored in the same directory as the chunks.

2. Encryption

To protect against unauthorized playback and copying, content can be encrypted. In this case, the encryption will be applied before writing to the storage.

At the moment, supported encryption standard is *ISO/IEC 23001-7: 2015 Part 7 (Common encryption in ISO base media file format files)*; the ISO BMFF/MP4 Fragmented container is supported. Any CENC-compatible DRM system can be used, for example Google Widevine or Microsoft PlayReady.

3. Content Playback

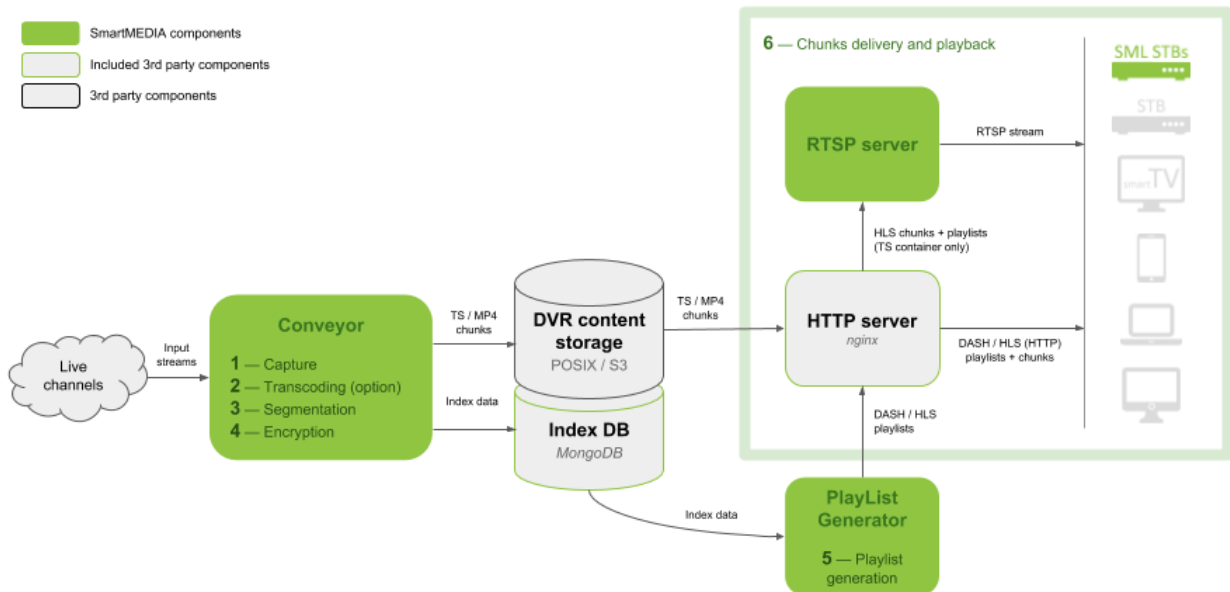
The subscriber device requests the playlists and chunks via the HTTP protocol and plays them. The solution includes an nginx HTTP server, which performs content delivery.

3. nDVR Services

nDVR Services

Unicast LiveTV and "network video recorder" (nDRV) services mean the retransmission of digital streams (for example, TV channels) with individual (unicast) delivery to subscriber devices over IP-networks. Examples of nDVR services include such services as TimeShifted TV (TSTV), PauseLive, or a network personal video recorder (nPVR).

To implement these services, streams have to be recorded onto storages and be delivered to the subscriber's device upon request. The content processing path from ingestion till delivery to the end-user by the SmartMEDIA components are shown in the following scheme:



nDVR Content Preparation Stages

1. Streams Injection

Incoming TV streams (multicast or unicast streams in the TS container) should be injected into SmartMEDIA Conveyor service.

Incoming streams must meet the following requirements:

- Container: SPTS MPEG2-TS (ISO/IEC 13818-1, ITU-T).
- Protocols:
 - UDP over IP Multicast (without RTP encapsulation);
 - HTTP (MPEG2-TS stream over HTTP);
 - HLS according to draft-pantos-http-live-streaming-05, without encryption.
- Supported video codecs: H.262/MPEG2, H.264/AVC, H.265/HEVC.
- Video resolution: up to 4K, up to 60fps. Both progressive and interlaced streams are supported.
- Supported audio codecs: AAC, AC3, MP2, MP3, DTS.

2. Transcoding

The SmartMEDIA allows you to perform transcoding of live input streams to multi profile adaptive bitrate, DASH or HLS, streams for OTT delivery. It runs on standard off-the-shelf hardware with Intel CPUs with QSV support or NVIDIA GPUs with NVENC support.

The transcoder function includes decoding, filtering (audio and video) and encoding. The incoming stream must be unencrypted (otherwise it can not be decoded).

The following filters can be applied to video streams:

- Deinterlacing (applied automatically if the incoming stream is interlaced);
- Resize;
- Changing the frame rate (fps).

Target formats can be:

- Video codecs: H.264/AVC, H.265/HEVC;

- Video format: up to 4K, up to 60fps, progressive scan;
- Audio codecs: AAC.

3. Segmentation, Indexing and Recording of Content

In order to provide nDVR services, as well as Live over HLS and DASH protocols, media streams must be recorded into the storage. To achieve the highest performance when delivering content to subscribers, SmartMEDIA records streams in the same form as they will be delivered to subscriber devices, all stream conversions (remultiplexing, encryption, etc.) are performed during recording.

SmartMEDIA supports 2 recording modes:

- **Main mode**, or mode with remultiplexing: the incoming stream is completely demultiplexed, only the desired tracks (audio and video) are used. Then the elementary stream is encrypted if necessary, packed into the desired container (MPEG2-TS or ISO BMFF) and written into the storage.

To work in this recording mode, the incoming stream should not be encrypted, otherwise all information about encryption will be lost and it will not be possible to reproduce it.

- **Pass-Through**: incoming TS-stream is written in the form as it comes to the server. The stream is not remultiplexed, it is divided into chunks and written into the storage. In particular, all timestamps are stored in the stream, as well as CC errors, if they were present in the incoming stream.

In this mode, the server can also accept encrypted TS streams if the TS packet structure is preserved and the NAL-unit headers are not encrypted (for example, DVB Simulcrypt or Common Encryption for MPEG2-TS encrypted streams).

Recording can be done both in POSIX-storages (local file systems, NFS, external storages), and in external object storage using the Amazon S3 protocol.

Simultaneously with the segmentation, the stream is being indexed. Based on the indexes HLS/DASH/Smooth Streaming playlists will be created later. Currently, the supported index storage is MongoDB.

It also should be considered that:

- if you're going to use adaptive bitrate streaming (ABR), all video streams should have the same GOP structure and have synchronous key frames (I-frames);
- the value of DTS counters for audio and video samples should not differ by more than 1.5 seconds;
- the value of DTS counters should increase monotonically;
- if the PMT table in the MPEG2-TS container was changed, a table version change is required;
- there should be no CC errors in the TS-stream.

4. Encryption

To protect against unauthorized viewing, copying, etc. during the recording process, the content can be encrypted according to one of the following standards:

- **HLS-AES** – only the MPEG2-TS container is supported and subsequent delivery over the HLS protocol. The entire chunk, including the headers of TS packets, is encrypted using the AES-CBC algorithm with PKCS#7 padding. Information about encryption can only be added to the HLS playlist (#EXT-X-KEY tag). As a DRM system, Verimatrix Adaptive CAS or another DRM with same API for obtaining encryption keys can be used.
- **ISO/IEC 23001-7: 2015 Part 7 (Common encryption in ISO base media file format files)**, abbreviated to CENC. The ISO BMFF/MP4 Fragmented container is supported. Only the elementary stream (payload) is encrypted, the container and the headers of the NAL packets of the video stream remain unencrypted. The data is encrypted using the AES-CTR algorithm. Any compatible system can be used as DRM, for example: Google Widevine or Microsoft PlayReady. The Widevine

Modular DRM API or SmartLabs UDRM API can be used for encryption keys retrieval.

5. Playlists Generation

The Playlist Generator component generates HLS or DASH playlists for the recorded content by end-user devices requests.

HLS playlists generation is supported for unencrypted content, as well as encrypted with the HLS-AES standard content recorded in the MPEG2-TS container.

DASH playlists generation is supported for content recorded in the ISO BMFF/MP4 Fragmented container.

6. Content Playback

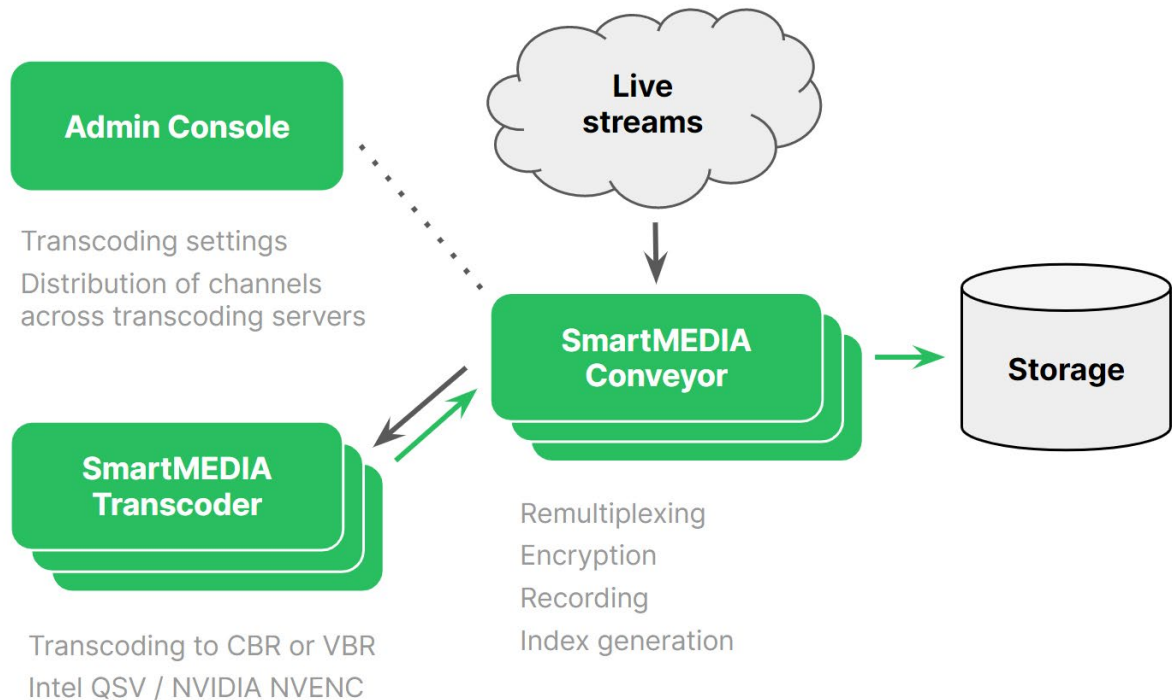
The subscriber device requests chunks for audio and video and plays them. The solution includes an nginx HTTP server, which performs the delivery of chunks over the HTTP protocol.

4. Transcoding

SmartMEDIA Transcoder

The **SmartLabs SmartMEDIA Transcoder** product allows you to perform transcoding of live input streams to multi profile adaptive bitrate, DASH or HLS, streams for OTT delivery. This software product runs on standard off-the-shelf hardware with Intel CPUs with QSV support or NVIDIA GPUs with NVENC support.

The solution architecture deploys two components of the SmartMEDIA Content Delivery Platform – *SmartMEDIA Conveyor* and *SmartMEDIA Transcoder*, which can be run on separate nodes and provide high stability and scalability. Transcoding profiles can be managed via a convenient web interface.



Key Advantages

- The solution allows you to use the most common generic hardware technologies for transcoding – Intel QSV and NVIDIA NVENC. You can reuse the existing hardware or opt for the new one. Wide range of **Intel CPUs** & **NVIDIA GPUs** is supported.
- Native integration with Intel and NVIDIA decoding/encoding API, unlike the ffmpeg-based solutions, ensures higher stability and lower resource utilization.
- Advanced algorithms avoid desynchronization of target streams and tracks, which often happens with ffmpeg-based solutions.
- *Transcoder* service can be shared between multiple *Conveyor* services, and vice versa. So you can build an optimal system configuration and, as a result, save on CAPEX & OPEX.
- High fault tolerance and easy scalability are the basis of the solution.
- Web-based GUI makes it easy to manage settings and channel distribution between transcoding servers.
- Native integration with **SmartTUBE** Service Delivery Platform.

QSV or NVENC

Taking into account factors such as hardware cost, power consumption, rack space, redundancy, and transcoding quality, we can conclude that the NVENC-based solution is much more cost effective in most cases. Tests of the **SmartLabs SmartMEDIA Transcoder** have shown that the price per transcoded channel in a NVENC-based solution can be up to 36% cheaper than in an QSV-based one. We present you a comparison of two indicative 6U installations:

- **QSV-based (non-redundant)**

1x 6U server based on Supermicro MBE-628E-820D MicroBlade Enclosure and CPU Intel Xeon E-2224G

vs.

- **NVENC-based (non-redundant)**

3x 2U servers built on Supermicro SYS-2029GP-TR Barebone and GPU NVIDIA RTX A2000

	Intel QSV	NVIDIA NVENC	Difference
Indicative HW price	\$ 61 414	\$ 25 371	> 2 times cheaper
Number of channels	280	180	36%
Channels per unit	47	30	36%
Price per unit	\$ 10 236	\$ 4 229	> 2 times cheaper
Price per channel	\$ 219	\$ 141	1/3 cheaper

In addition, as of June 2022, the average delivery time of the QSV-based solution is 2–4 months, while the NVENC-based one – 1 month.

All examples of prices and delivery times above are indicative, and the company is not responsible for their accuracy at the time of reading.

Component Overview

SmartMEDIA Conveyor

The *SmartMEDIA Conveyor* service is responsible for remultiplexing, segmenting, indexing and recording LiveTV streams to the storage.

Content can be delivered to the *SmartMEDIA Conveyor* server within the MPEG2-TS container using one of the following protocols:

- UDP over IP Multicast (without encapsulation in RTP),
- HTTP Stream,
- HLS (according to draft-pantos-http-live-streaming-07), without encryption.

SmartMEDIA Conveyor performs:

- remultiplexing,
- media data encryption (if necessary),
- recording to the storage,
- index generation and writing indexes to the database (MongoDB).

Adaptive Streaming Support

To record several streams of the single channel that have different bit rates (a.k.a. Adaptive Streaming), *SmartMEDIA Conveyor* synchronizes them at the recording stage and creates one index record for all of the chunks of different bitrates of the same time interval.

SmartMEDIA Transcoder

SmartMEDIA Transcoder service receives a stream of media samples from the *Conveyor* over the TCP connection, transcodes the stream and sends the resulting stream (or multiple streams in case of ABR) back to the *Conveyor*. It can be started on the same host where *Conveyor* runs as well as on the other, "remote" host and work over the TCP/IP network.

SmartMEDIA Transcoder service uses proprietary protocol to exchange the data with *Conveyor* and cannot receive streams directly.

5. Multicast Content Protection

Multicast Content Protection

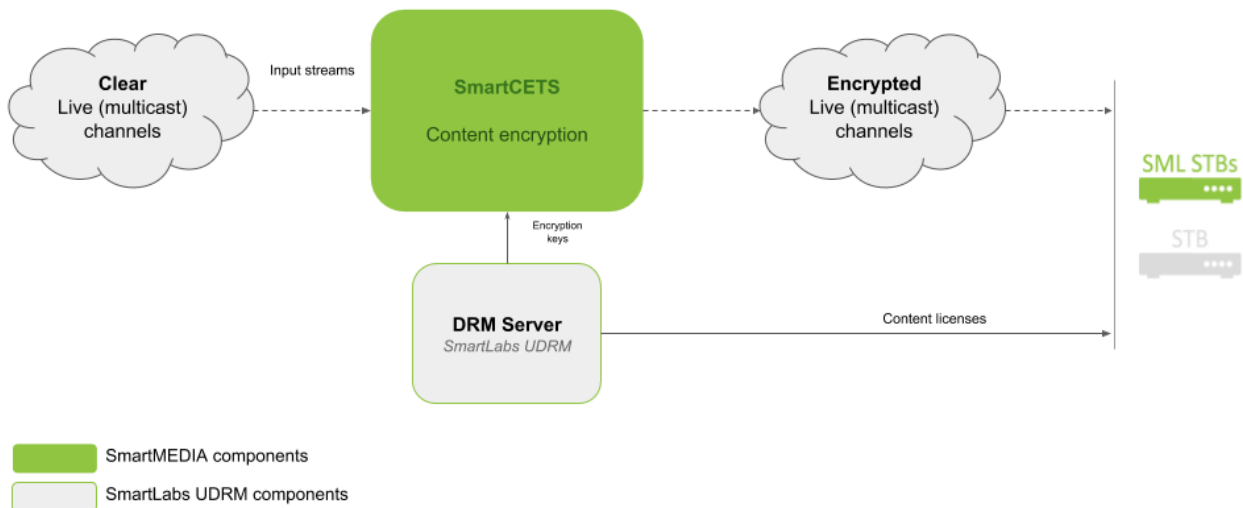
SmartMEDIA allows you to encrypt MPEG-TS content in real time according to the *ISO/IEC 23001-9 (Common encryption of MPEG-2 transport streams)* standard. The SmartCETS service allows you to encrypt multiple multicast streams simultaneously; the cets utility can be used to encrypt a single stream or MPEG2-TS files.

After receiving the list of input multicast streams, SmartCETS/cets requests keys for encryption from the key server over the SmartLabs UDRM protocol. Encrypted content is either written to a file or broadcast to another multicast group using the UDP protocol. For the SmartCETS service, the list of input streams for encryption along with other parameters must be specified in the configuration file. For the cets utility, all parameters must be passed as command-line arguments.

The following formats are supported:

- Video: H.262/MPEG2, H.264/AVC, H.265/HEVC;
- Audio: AAC/ADTS, AC3, DTS.

The utility also stores information about its work in the log, e.g. obtaining keys, stream errors, keys retrieval errors, etc.



Features of the Encryption Algorithm

Encryption is performed according to the *ISO/IEC 23001-9 (Common encryption of MPEG-2 transport streams)* standard, namely:

- the original MPEG-TS container is saved (except PMT), unknown tracks are not encrypted;
- in the PMT of the source stream, the descriptors described in ISO/IEC 23001-9 are added for each encrypted track;
- the payload of each video/audio track found is encrypted;
- the utility complies with ISO/IEC 23001-9 recommendations regarding the H.264/H.265 and AAC payload encryption (i.e., the payload of TS packets containing VPS/SPS/PPS/SliceHeader and ADTS Fixed header is not encrypted);
- for H.262, AC3 and DTS, the payload of TS packets containing PES-headers (*Packetized Elementary Stream*) is not encrypted;
- before each PES header, packets containing PSSH received from the key server for that track are inserted into the stream;
- before each PES header, ECM (*Entitlement Control Messages*) with content corresponding to ISO/IEC 23001-9, pertaining to this PES, is inserted into the stream. If the ECM content does not fit into the payload of one TS packet, the next

ECM packet will be inserted where its contents begin to act (and not at the beginning of the next PES packet).

6. Load Balancing and Failover

SCALING AND FAULT TOLERANCE WITHIN A SINGLE SITE

Can be implemented by:

- using external fault-tolerant or clustered storage systems (POSIX and object storages);
- reserving SmartMEDIA Conveyor services in "1 + 1" mode (Active/Standby);
- reserving services for playlists generation in "N + M" mode (Active/Active).

GEOGRAPHICALLY DISTRIBUTED STRUCTURE (CDN)

SmartMEDIA Redirector allows you to implement a geographically distributed solution for the media content delivery.

Functionality

- HTTP (for DASH and HLS) and RTSP requests balancing.
- Grouping the servers into logical farms.
- Providing different balancing policies between the servers of the farm.
- Setting priorities for specific servers.
- Servers failover in the group.
- Redundancy of server groups.
- Caching information about the content availability on a specific server.

Balancing Criteria

- IP-address/subnet of the subscriber — for specified subnets, the viewing order of server groups is assigned.

- Availability of content on servers – farms are polled from the "best choice" for specific subnet till the backup one; if there is no content on all servers in the farm, the servers of the next farm are polled, etc.
- When balancing requests between group servers, the availability of content on servers and the weight of servers are taken into account.