



# SmartLabs

SmartMEDIA 5.0

Functionality Overview



# Contents

<b>1. Review of SmartMEDIA</b>	<b>3</b>
1.1 Technology Used	3
<b>2. Key Features of SmartMEDIA</b>	<b>3</b>
<b>3. SmartMEDIA Functions</b>	<b>3</b>
3.2 VoD Services	3
3.2.1 VoD Content Preparation Stages	4
I. Segmenting and Playlist Generation	4
II. Encryption	4
III. Content Playback	5
3.3 nDVR Services	5
3.3.1 nDVR Content Preparation Stages	5
I. Streams Injection	5
II. Transcoding	5
III. Segmentation, Indexing and Recording of Content	6
IV. Encryption	7
V. Playlists Generation	7
VI. Content Playback	8
3.4 Multicast Content Protection	8
<b>4. Load Balancing and Failover</b>	<b>9</b>
4.1 Scaling and Fault Tolerance Within a Single Site	9
4.2 Geographically Distributed Structure (CDN)	9
<b>5. Terms, Abbreviations, Notations</b>	<b>10</b>



# 1. 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, Apple iOS devices, Apple tvOS, Android, Microsoft Windows Mobile/Windows Phone, Adobe Flash Player add-ons, Microsoft Silverlight Player, 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.

## 1.1 Technology Used

- *Programming language:* C/C++, JavaScript/HTML (for Admin UI).
- *Third-party components, frameworks, etc.:* openssl, ffmpeg (--non-free), libzvbi, openjpeg, MongoDB.

# 2. 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.

# 3. SmartMEDIA Functions

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.



## 3.2 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:

### 3.2.1 VoD Content Preparation Stages

#### *I. 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.

## II. 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.

## III. 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.

# 3.3 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:

### 3.3.1 nDVR Content Preparation Stages

#### I. 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.

## *II. Transcoding*

The transcoding of Live-streams is a new feature introduced in SmartMEDIA version 5.0.

This functionality is implemented using Intel codecs that are part of the Intel Media SDK, and requires the use of compatible processors (CPUs) and chipsets. For more information on hardware requirements, see the document "SmartMEDIA 5. Administrator's Guide" and the Intel website (<https://software.intel.com/en-us/media-sdk>).

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.

## *III. 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 thread.

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.

#### IV. 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.

### V. 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.

### VI. 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.

## 3.4 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).

## 4. Load Balancing and Failover

### 4.1 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).

### 4.2 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.

## **5. Terms, Abbreviations, Notations**

### **Stream decoding**

Obtaining an uncompressed stream (a sequence of uncompressed frames for video or PCM stream for audio) from a compressed (encoded) stream.

### **Demultiplexing**

The process of obtaining elementary streams (audio, video and text) from media container.

### **Stream encoding**

Compression of audio or video stream using certain algorithms (codecs).

### **Media container**

Media container (or *container*) — a format for storing video, audio and text data with synchronization marks between them. Several media streams of the same type can be stored in the media container, e.g. a movie (video stream) with several audio tracks (audio streams) and subtitles (text streams). The media container determines the size and structure of the attached data, but does not specify the encoding standard. Some of the most popular media containers today are MPEG-TS, MP4, MKV, MOV, AVI, 3gp, etc.



## Multiplexing

Multiplexing (or *muxing*) — an operation or process of packaging elementary video and audio streams into a single media container with time synchronization.

## Playlist

A file containing the data needed to play content on the subscriber device (including links to chunks). For more information, see:

- <https://tools.ietf.org/html/rfc8216#page-6> (for HLS),
- <https://www.iso.org/standard/65274.html> (for DASH).

## Manifest

see Playlist. The term is used for the DASH protocol.

## Remultiplexing

The process or operation of changing one media container to another one without elementary streams transcoding.

## Transcoding

The process or operation of changing the codec (compression algorithm) of the audio or video stream.

## Chunks

Content segments referenced by the playlist.

## Elementary stream

The stream generated by the audio or video encoder. Contains only one kind of data — video, audio, subtitles, PCR or others.

## CC

*Continuity Counter* — counts the packets with the same PID in the TS stream. Allows the decoder to detect packet loss. It can not be an unambiguous sign of packet loss, because the counter size (4 bits) allows storing only 16 values (from 0 to 15) and the probability of losing the number of packets, multiples of 16, is high enough. At the same time, CC violations can occur in case of incorrect merging several TS streams.

**CDN**

*Content Delivery Network* — serves to optimize the unicast traffic delivery in the operator's network or over the Internet.

**DRM**

*Digital Rights Management* — software or soft hardware restricting or preventing unauthorized access (copying, modification, viewing, etc.) to the content, or allowing to track such actions. DRM is a set of access control systems.

**DTS**

*Decoding Time Stamp* is used to indicate the order of frames decoding. Typically, frames in TS and MP4 containers are arranged in the order of DTS increasing.

**DVR**

*Digital Video Recording* — the possibility of digital TV-broadcasting, meaning recording and storing incoming streams (video, audio, text, metadata). Can be implemented both on the subscriber device (LocalDVR) and on the service provider equipment (Network DVR, nDVR). In particular, nDVR allows to implement such digital TV services as nPVR and Pause Live.

**nPVR**

*Network Personal Video Recorder* — a digital TV service allowing the subscriber to record current broadcast and watch it later. The recording is performed on the service provider site. See also DVR.

**Pause Live**

The digital TV service allowing the subscriber to pause the current broadcast and resume playback later. The maximum broadcast suspension time may be limited by the service provider.

**PID**

*Packet Identifier* — the identifier of the packet in the TS stream. Packets containing fragments of the same elementary stream have the same PID by which they are recognized by the decoder.

**PMT**

*Program Map Table* — the program structure table in the TS stream. PMT contains a list of all elementary streams of the program with their PIDs. SmartMEDIA supports TS streams with only one PMT (SPTS).

**PTS**

*Presentation Time Stamp* is used to specify the sequence of frames appearing on the screen. May differ from the decoding sequence in order to optimize the stream compression.

**S3**

*Amazon Simple Storage Service* is an online web service provided by Amazon Web Services. For more information, see <https://aws.amazon.com/en/s3/details/>.

**TSTV**

*Time Shifted TV* — a digital TV service allowing the subscriber to playback finished broadcasts.

**VoD**

*Video on Demand* — a service of personal delivery of movies, series and TV program archives over the digital cable, satellite or terrestrial TV network.