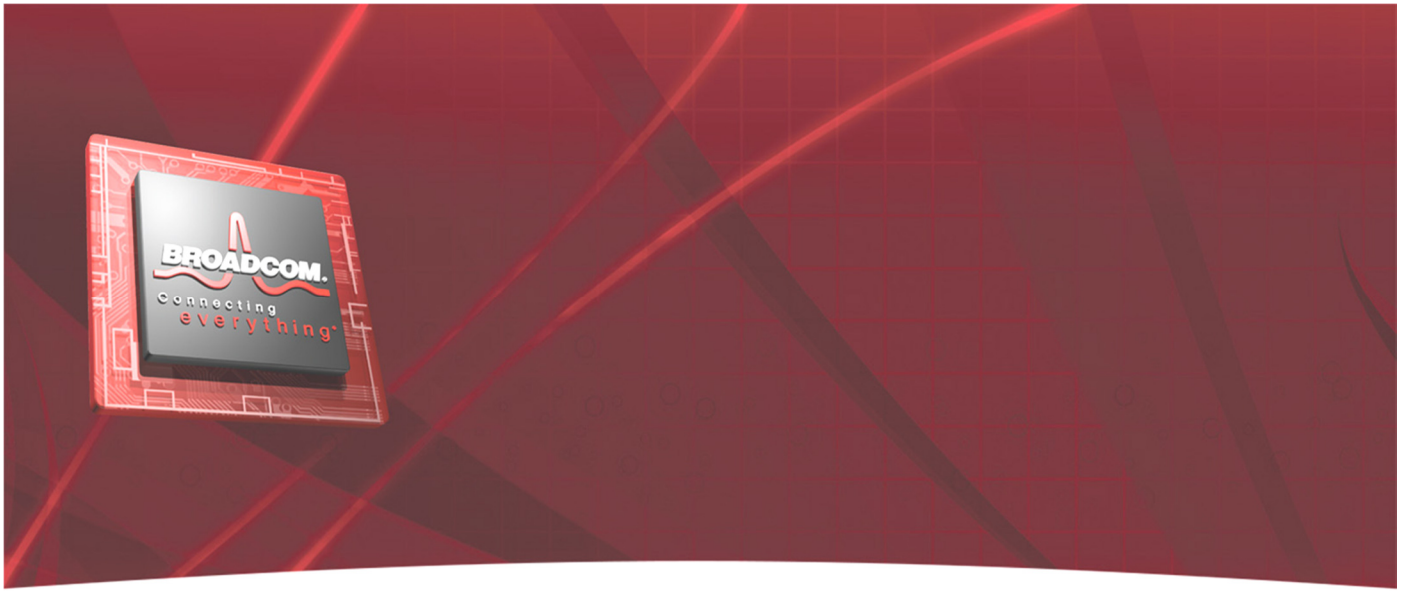


OCAP DVR EXTENSIONS LIBRARY

REVISION 1.4

Broadcom Corporation
5300 California Avenue
Irvine, California, USA 92677
Phone: 949-926-5000
Fax: 949-926-5203

Web: www.broadcom.com

TABLE OF CONTENTS

REVISION HISTORY

Revision	Date	Change Description
1.0	03/24/2010	Initial draft
1.1	03/24/2011	Add DRM Service
1.2	04/11/2011	Add Media File I/F and Data Injection Service
1.3	04/05/2012	Add Sequence Diagrams
1.4	02/27/2013	Add DVR storage description

1. OVERVIEW OF DVR EXTENSION LIBRARY

DVR extension library is a nexus application library placed right on top of the Nexus API. The highlight of the library is the segmented recording which prevents disk fragmentation irrespective of how long the disk has been in use in SMS/VMS platforms and TSB conversion by copying the meta-data file entries rather than the actual media files.

1.1 ARCHITECTURE AND PERFORMANCE

- Fragmentation free storage management through segmented file system
- Support TSB and efficient TSB conversion
- High performance with low CPU usage [multiple recordings/playback/streaming]

1.2 FEATURES

- Media probing from encrypted recordings used during streaming and transcode operations
- Recording specific custom PAT/PMT injection used in HN streaming
- HN file IO interface with trick mode and flow control
- Seamless record overflow recovery used during service interruption
- Encryption during recording [M2M/CPS]
- Decryption during playback and HN streaming [M2M]
- Manage vendor specific information in the metadata
- Efficient program listing through metadata grouping
- Reservation of TSB space [per tuner context]

1.3 FLEXIBILITY

- SW library could be leveraged across different types of platforms [Cable/Sat/IP]
- Encapsulate complex nexus API sequence into simple API sequence [Transcode is an example]
- On demand resource management
- No conflict with non-DVR resources.

2. SEGMENTED FILE SYSTEM

2.1 DVR STORAGE VOLUME DEFINITIONS

- **APPLICATION INPUT**
  - MS = MEDIA FILE SEGMENT SIZE
  - FR = HIGHEST VIDEO FRAME RATE
  - BR = WORST CASE BIT RATE
  - START SECTOR/BLOCK
  - SIZE OF SEGMENTED DVR VOLUME [SVS]
- **MEDIA STORAGE CALCULATION**
  - NS = NAVIGATION FILE SEGMENT SIZE = MS/(BR/8) * FR * SIZE OF MAX NAVIGATION ENTRY
  - SVS = NS*X + MS*X + (MS/100)*X [CALCULATE X]
  - MPS = MEDIA PARTITION SIZE = MS*X
  - NPS = NAVIGATION PARTITION SIZE = NS*X
  - MDPS =META-DATA PARTITION SIZE = (MS/100)*X
  - X = NUMBER OF MEDIA & NAVIGATION FILE SEGMENTS PRE-ALLOCATED USING FALLOC/FTRUNC

2.2 DVR STORAGE VOLUME

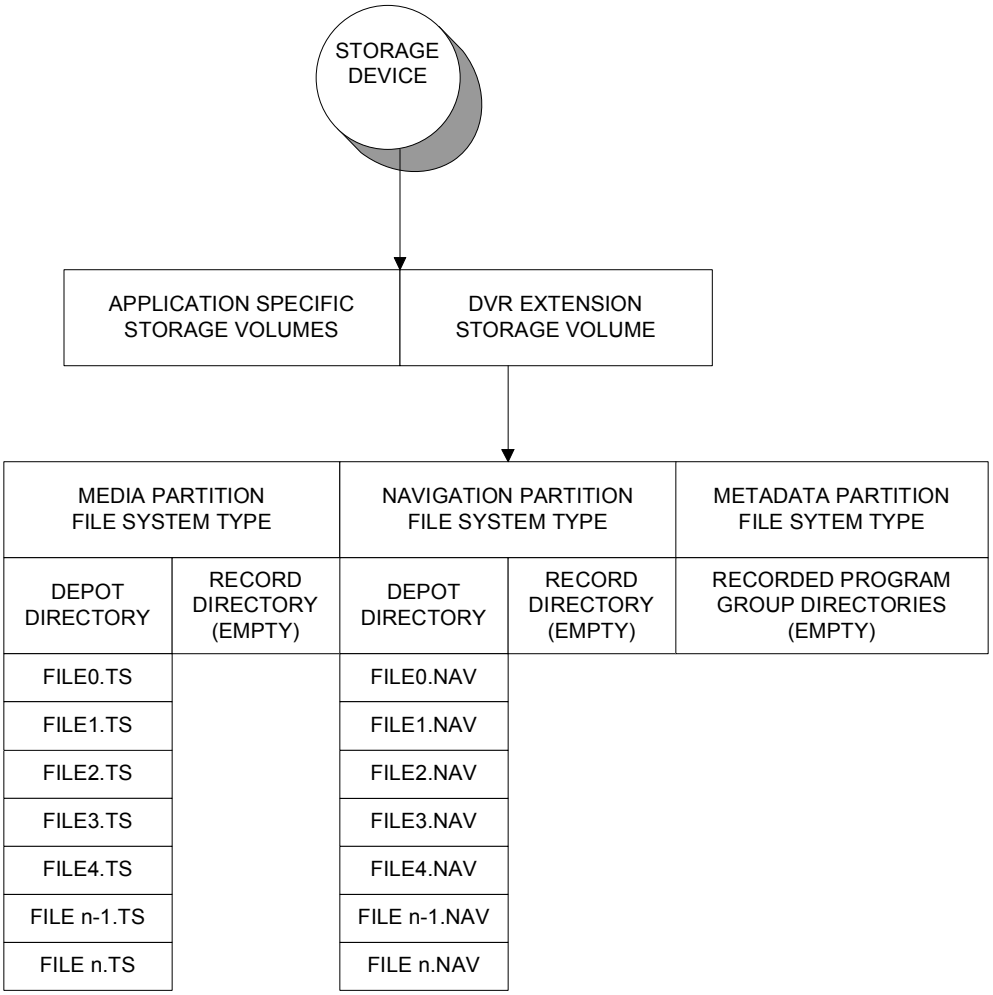


FIGURE-1. DVR STORAGE VOLUME

As shown in the FIGURE-1, DVR library creates a DVR storage volume based on the application provided volume block start and volume size parameters. The DVR storage volume is divided into 3 partitions:

- **MEDIA PARTITION**: Media partition can be formatted to have any file system type, but the default file system chosen is EXT4. “depot” and “record” directories are created by default in this partition after formatting the partition with EXT4. Initially, “record” directory would be empty and “depot” directory would have “n” number fixed sized files created through FTRUNC or FALLOC.
  By default, the DVR library assumes the media file segment size to be 150MB.
  When some of the media file segments are used in an ongoing recording or are a part of permanent recording, the media file segments are moved from “depot” to “record” directory and when a segmented recording is deleted, some of the media file segments are moved from the “record” to “depot” directory.

- **NAVIGATION PARTITION:** Navigation partition can be formatted to have any file system type, but the default file system chosen is EXT4. Like media partition, navigation partition also has “depot” and “record” directories. Initially, “record” directory also is empty and “depot” directory has “n” number of files of fixed size created through FTRUNC/FALLOC.
  Each navigation file segment is associated with a media file segment in a recording and the navigation entries in a nav file segment correspond to data only in the associated media file segment. Fixed size of the navigation file segment is dependent on the application passed in fixed size of media file segment. When some of the navigation file segments are used in an ongoing recording or are a part of permanent recording, the navigation file segments are moved from “depot” to “record” directory and when a segmented recording is deleted, some of the navigation file segments are moved from the “record” to “depot” directory.

- **META DATA PARTITION:** Meta Data partition can also be formatted to have any file system type, but the default file system chosen is EXT4. This partition would have 3 meta data files per recording
  - Media Meta Data File
  - Navigation Meta Data File
  - Media Info File

  The recordings can be grouped into multiple groups by creating sub directories in this partition.

2.3 SEGMENTED RECORDING FILE LAYOUT

Assume that a recording named PROGRAM1 grouped under GROUP1 is recorded on the DVR volume. As shown in the FIGURE-2, PROGRAM1 would be associated with 3 meta-data files created under GROUP1 subdirectory in the METADATA partition –

- PROGRAM1.TS: PROGRAM1.TS file has a list of FILE NODES which describe the parameters of the media file segments associated with the PROGRAM1 recording.
- PROGRAM1.NAV: PROGRAM1.NAV file also has a list of FILE NODES which describe the parameters of the navigation file segments associated with the PROGRAM1 recording.
- PROGRAM1.INFO: PROGRAM1.INFO has 2 data sections –
  - FIX SIZED DATA SECTION: is encapsulated using a MEDIA NODE data structure. MEDIA NODE has information on all the files in METADATA partition associated with the PROGRAM1 recording.
  - VARIABLE SIZED DATA SECTION: holds VENDOR SPECIFIC DATA.

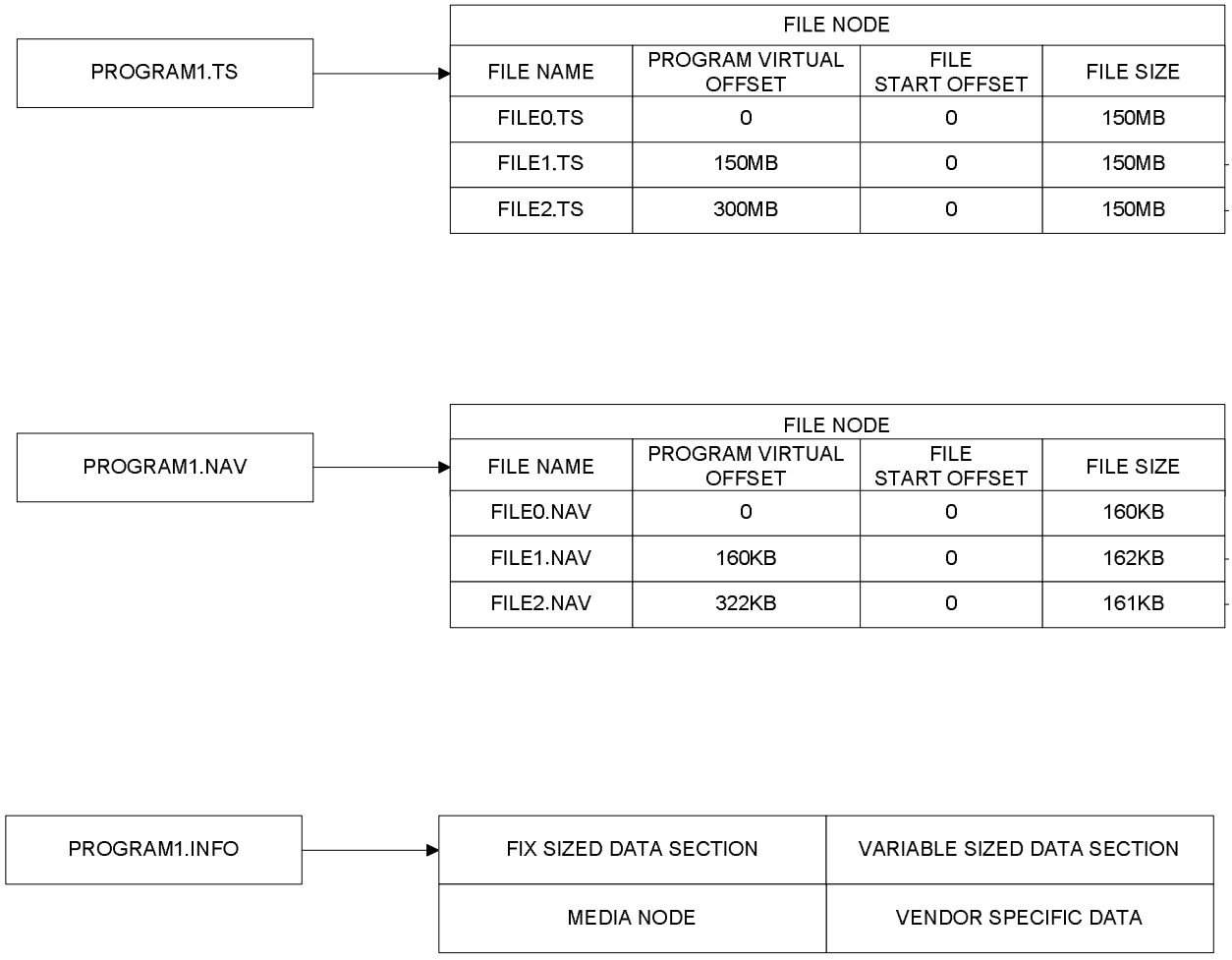


FIGURE-2. SEGMENTED RECORDING META-DATA

MEDIA NODE as shown in FIGURE-3 is a fix sized binary data section in PROGRAM1.INFO and has all the information about a recording.

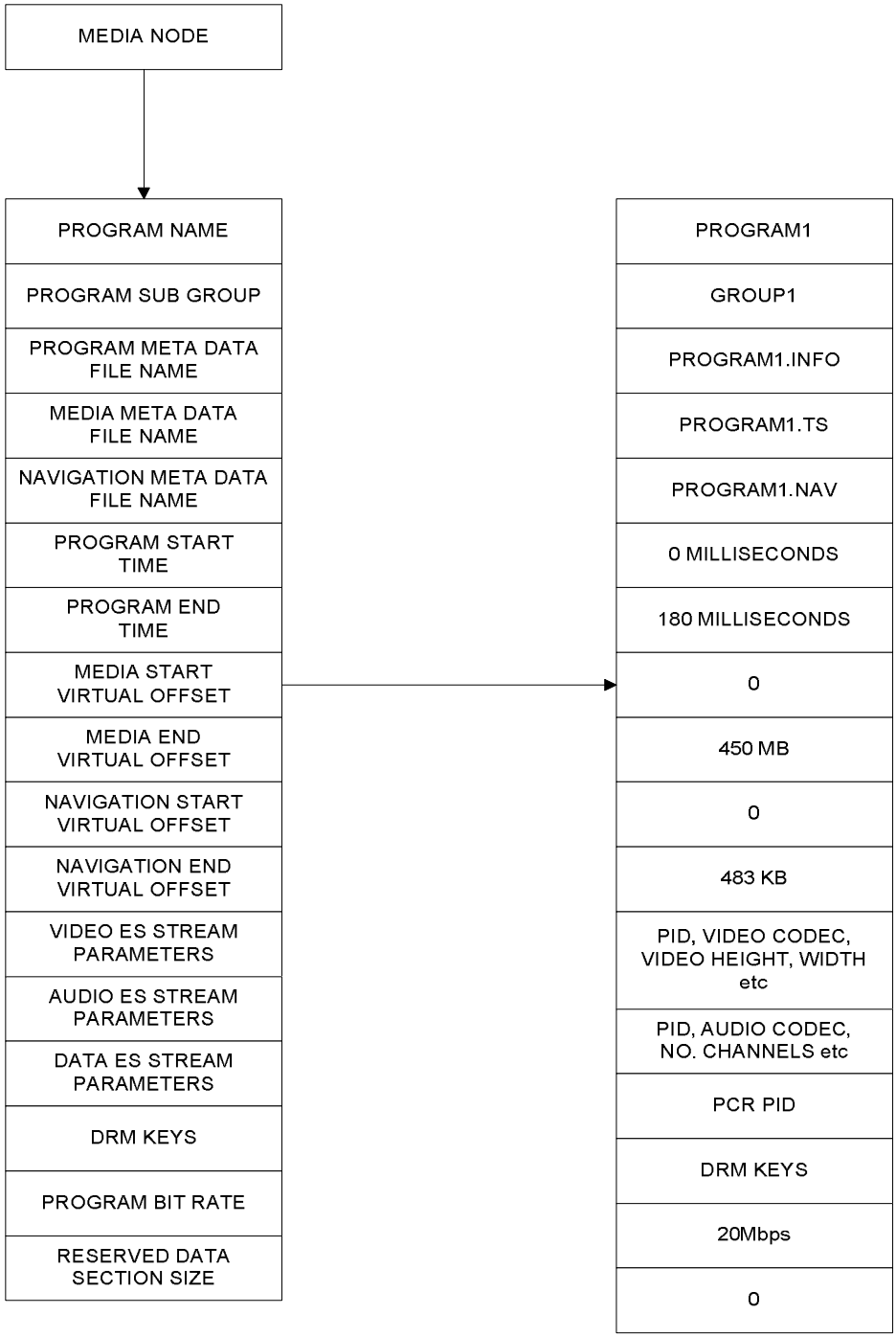


FIGURE-3. MEDIA NODE

3. DVR EXTENSION LIBRARY SOFTWARE

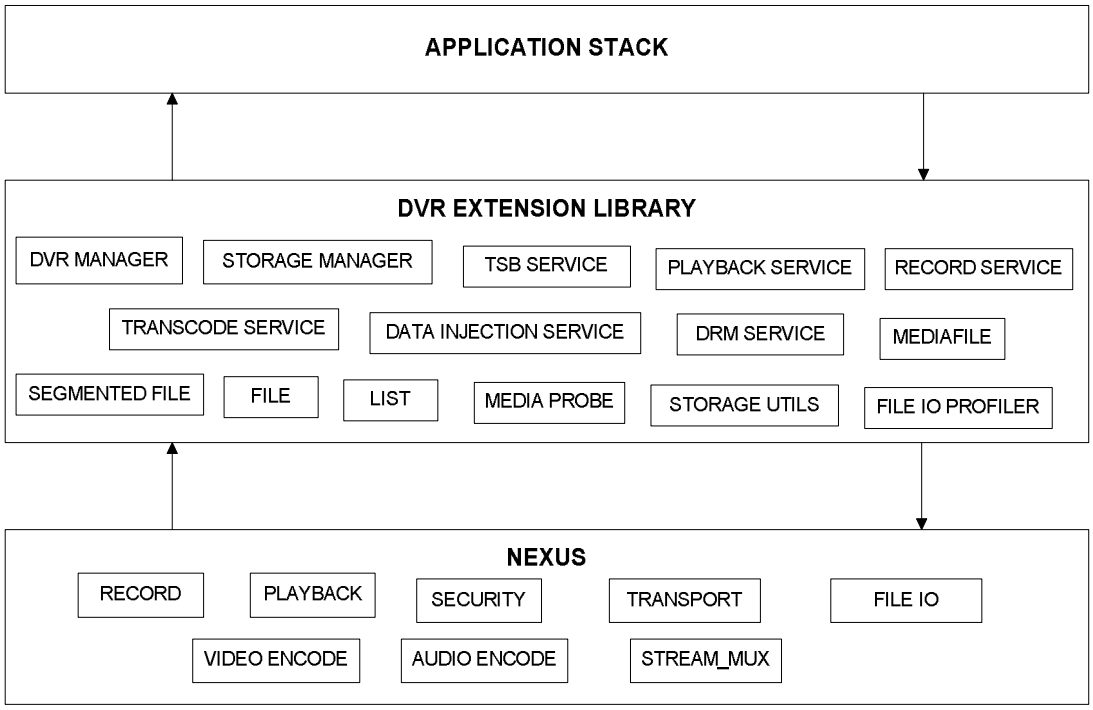


FIGURE-4. DVR EXTENSION SW MODULES

The library comprises of several sub-modules which are categorized as managers, services and utilities.

Sub-Modules

- Manager – DVR Manager, Storage Manager
- Services – TSB Service , Playback Service, Record Service, Trans-code Service, DRM Service, Data Injection Service.
- Utilities – Media File Interface, Segmented File, File, List, Media Probe, Storage Utilities, File IO profiler.

3.1 NEXUS MODULES IN DVR EXTENSION LIBRARY

DVR extension library uses several nexus modules which are briefly described in the following section.

3.1.1 NEXUS RECORD

As shown in FIGURE-5, in the Nexus Record module, MPEG2 transport media is taken from any of the available input sources and stored on the storage device for later viewing. Along with the MPEG2 transport media, the start code detect hardware in the transport processor generates an index table buffer having start code entries, which serves as a simple index into the MPEG2 transport data. This index consists of an entry for every start code in the MPEG2 transport media. Each start code index entry points to either the beginning of a new picture, a sequence header, or a PTS entry etc. These raw start code entries are processed by Broadcom navigation indexer to generate a navigation table for the MPEG2 transport stream. The navigation table has one entry per video frame in a recording. The host processor is responsible for the transfer of all the data recorded in the compressed data and indexed data FIFOs to the storage device in a timely manner to avoid overflowing of the FIFOs. More details on nexus record module are available in the transport HW description and nexus architecture documents.

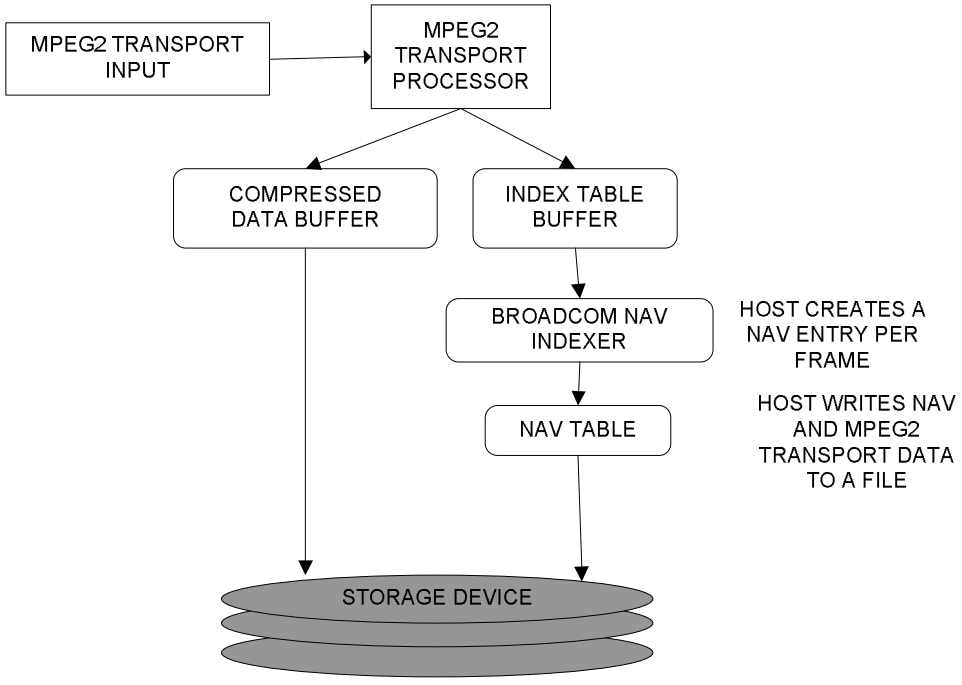


FIGURE-5. NEXUS RECORD

3.1.2 NEXUS PLAYBACK

As shown in FIGURE-6, in the Nexus Playback module, the host processor uses the NAV table to determine what MPEG data should be read from the storage device and sent to the video decoder. In the case of normal playback, the starting point in the MPEG file is determined, and then all MPEG data thereafter is sent. However for host based DVR operations, the NAV table must be constantly utilized in order to determine what MPEG data should be skipped and what should be sent to the decoder. Also, the content rating and encryption information must be sent to appropriate system components at the correct time. Decoder based DVR operations are performed based on the information in the Index Table Buffers and navigation table entries are not utilized. STC trick modes neither use the navigation entries nor the start code entries; rather the processing speed of the decoder is increased or decreased by manipulating the system time clock fed to the decoder. More details on the nexus playback are available in Transport HW description and nexus architecture documents.

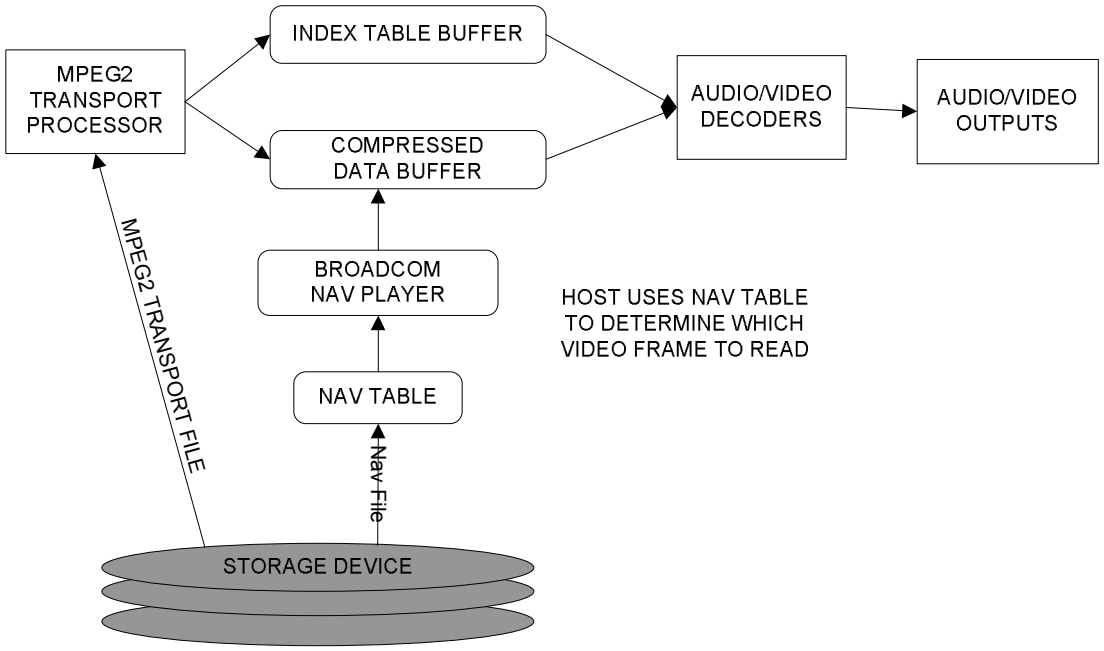


FIGURE-6. NEXUS PLAYBACK

3.1.3 NEXUS TRANSCODE

As shown in FIGURE-7, the nexus trans-coding module can have either live input stream or recorded stream as an input. The main components of nexus trans-code module are video encoder, audio encoder, stream multiplexer and audio multiplexer. The input of video encoder is a sequence of de-compressed video frames passed from the video decoder and display modules. The output of the video encoder is passed into a stream multiplexer module whose output is added into a record context using a play pump PID channel. Audio content from either live input or a recorded file can be either bypassed and added directly into a record context or is encoded into a different audio codec and added to a record context. If the input audio content is bypassed, then audio encoder is bypassed and the decode audio content is directly sent to an audio multiplexer which is added as an input to a stream multiplexer whose output is again added to a record context though a play pump PID channel. If the audio content is encoded, then the audio content is decoded using an audio decoder and the decoded content is sent an audio encoder. Audio encoder output is sent to an audio multiplexer. The output of the audio multiplexer is added to a record context through a play pump PID channel. Multiple audio contents can be trans-coded or bypassed and added to a recording context. The synchronization of audio and video is done through STC channels and play pump pacing. System data transport packets can also be inserted into a trans-coded output in a recording context. More details on nexus trans-code are available in nexus trans-code document.

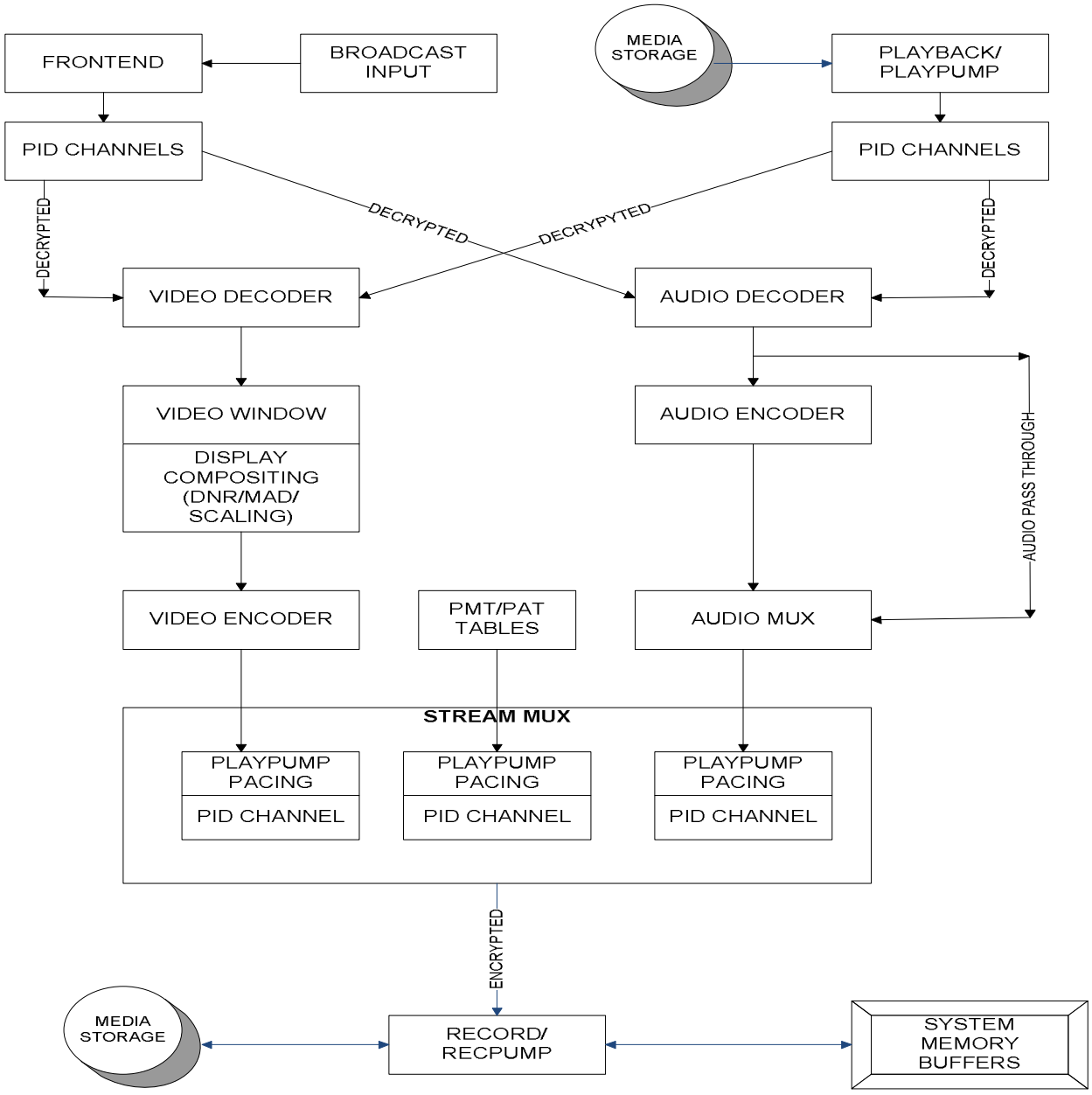


FIGURE-7. NEXUS TRANSCODE

3.1.4 NEXUS PACKET SUBSTITUTION

As shown in FIGURE-8, the packet substitution module can be used to insert transport packets like PAT/PMT packets in an ongoing recording. Generally packet insertion rate is less than 1 Mbps per recording. Host can send the transport packets into a PSUB buffer. There could be multiple PSUB channels to allow packet insertion in multiple recordings simultaneously. Each PSUB channel needs a PID channel to insert the transport packets from the PSUB buffer into a recording.

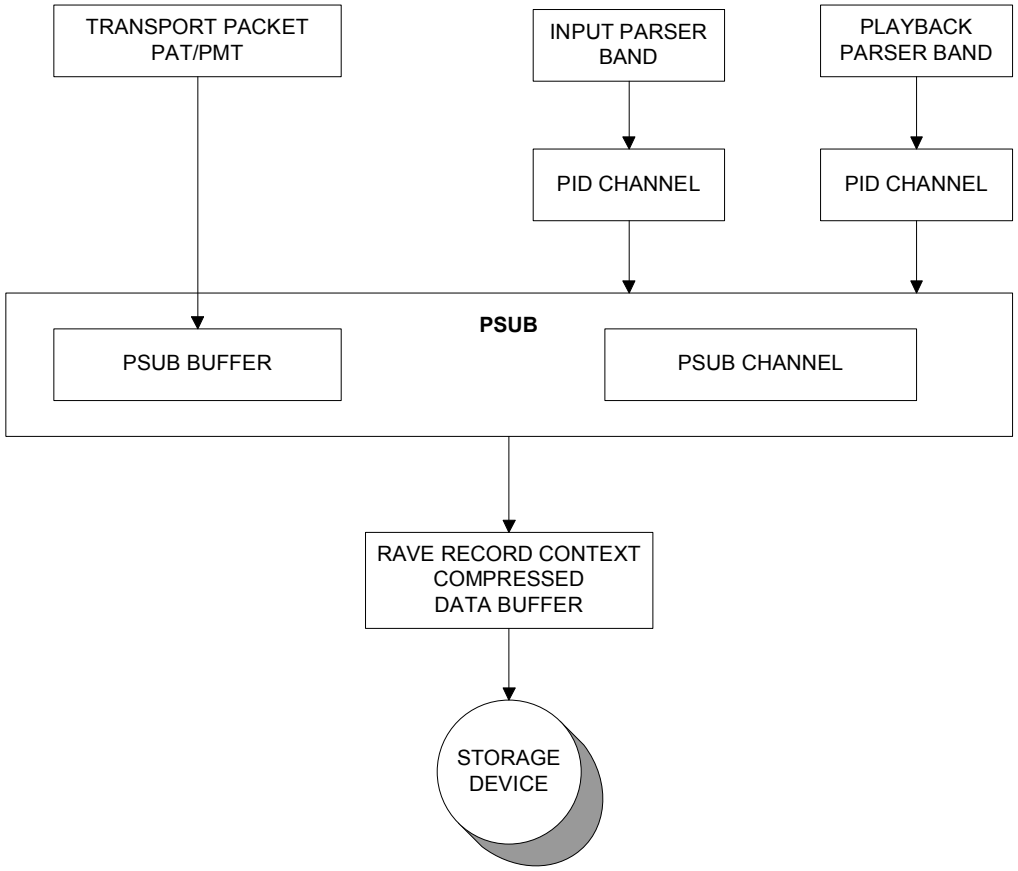


FIGURE-8. NEXUS PACKET SUBSTITUTION

3.1.5 NEXUS SECURITY

As shown in FIGURE-9, the application calls into Nexus Security which in turn calls into HSM PI. The HSM PI then sends corresponding BSP (Broadcom Security Processor) commands to the BSP

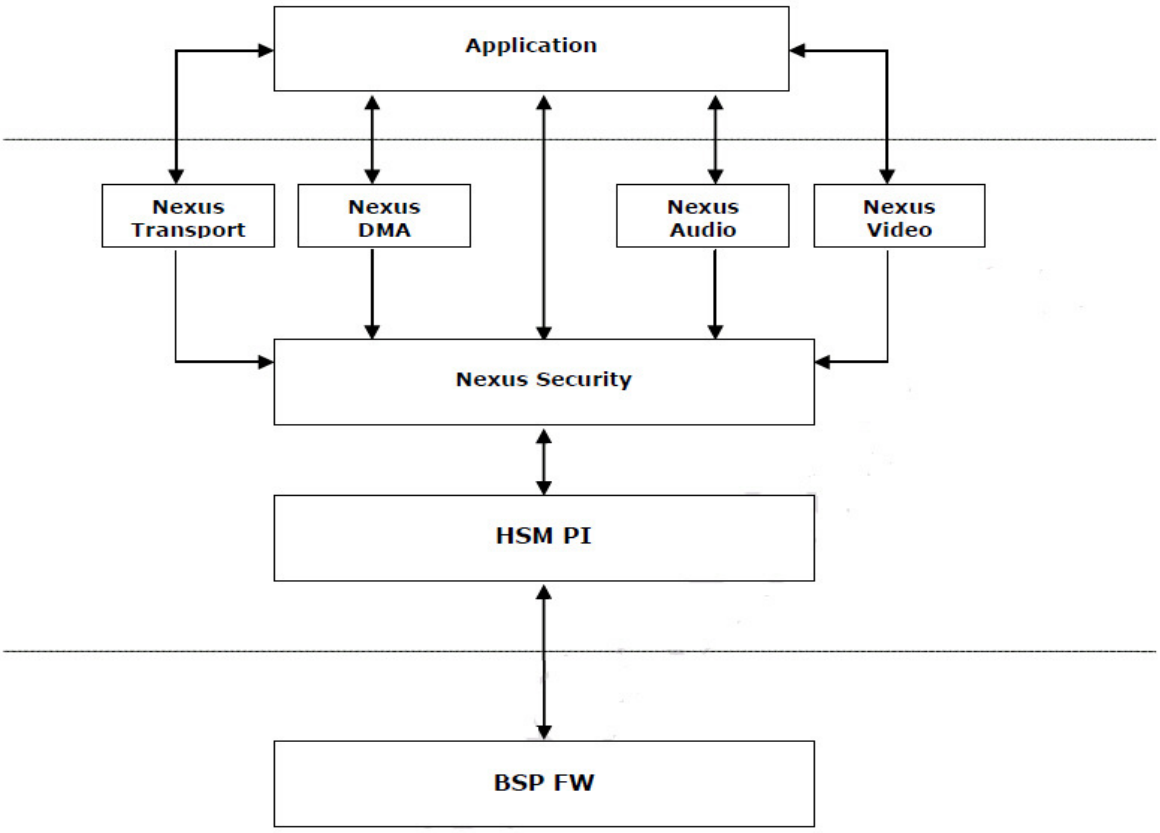


FIGURE-9. NEXUS SECURITY

The Nexus Security Module has four upper modules, namely, Nexus Transport, Nexus DMA, Nexus Audio and Nexus Video modules. Those upper modules may need to call into the security module for certain applications. The security functionalities are exposed via the Nexus Security Module which is built on top of the HSM PI. Encryption and decryption of recording can be done through either a CPS/CPD (Content Protection Scrambler/Content Protection Descrambler) or a DMA engine. Encryption and decryption of recordings are done by associating the PID channels with a key slot. More details on nexus security are available in the nexus security document.

3.2 DVR EXTENSION LIBRARY MODULES

3.2.1 STORAGE MANAGER

A media storage dvr volume consists of 3 physical partitions. Only 1 dvr volume can exist in a storage device, and up to 4 dvr volumes can be connected at the same time. Storage manager provides interfaces to create needed partitions format and mount media storage dvr volume to the user. It also provides interfaces to allocate and free segments to other dvr extension library services.

When storage manager is initialized, it scans connected storage devices for valid existing media storage dvr volume and registers if found. Hot plugged storage devices can also be registered during runtime.

It provides following functionalities –

- Creates a media storage dvr volume in a storage device. It will create partitions needed. It supports GUID partition only. MBR is not supported. After creation, it is ready to format.
- Formats a media storage dvr volume. It will format each media storage partitions and create required folders and segment files in media and nav partition.
- Mounts a media storage dvr volume. It will mount media, nav and metadata partitions to corresponding directories with a same mount name. When multiple volumes are connected and mounted, mount name is given automatically starting from 0. Mount name is given as /mnt/dvr<number>. For example, first dvr volume will have mount name as /mnt/dvr0, and partitions will be mounted at /mnt/dvr0-media,/mnt/dvr0-nav and /mnt/dvr0-metadata correspondingly.
- Unmounts a media storage dvr volume. I will unmounts partitions from corresponding mount directory and free its mount name.
- Register a new hard disk. It will check and register a new hard disk if it has valid media storage dvr volume. If it is valid, it will be ready for mount. It can be called when a new hard disk is connected to the system.
- Unregister a hard disk. It will unregister a dvr volume. It can be called when a hard disk is disconnected.

3.2.2 DVR MANAGER

DVR Manager holds information about all the services' and interfaces' instances and must be initialized after Storage Manager and before any other library modules are used by the application stack. There can be only one instance of a DVR Manager module.

It provides the following functionalities –

- Creates a scheduler which is used by other services and interfaces for handling asynchronous events and timer based activities.
- Since data injection requires HW PID channels that aren't associated with any of the application provided A/V and data PIDs, it allocates HW PID channels which are allocated based on DOCSIS reserved PIDs. This is to ensure that MPEG2 Transport stream PIDs don't conflict with the data injection PIDs.
- Move a segmented recording from one subgroup to another subgroup in the DVR storage volume.
- Delete a segmented recording from the DVR Storage volume.
- Pre-allocate the segmented files for a recording. This is used to pre-allocate the file segments for TSB contexts.

3.2.3 DATA INJECTION SERVICE

Data Injection Service encapsulates the nexus packet substitution module. Each data injection service instance uses a HW PID channel and PSUB channel and can be associated with a record service or TSB service instance for inserting application provided PAT/PMT transport packets into an ongoing recording.

HW PID channels used for data injection service instances are statically allocated without having them associated with any of the app provided audio/video/data PID values. The maximum data injection service instances allowed would be equal to the number of PSUB HW channels available in a HW platform.

3.2.4 DRM SERVICE

DRM Service encapsulates the nexus security module. Each DRM Service instance would be associated with a security key slot and each key slot can be associated with one or more PID channels. DRM service instance can be associated with a TSB Service or a Playback Service or a Media File Interface (HN Streaming) for encrypting or decrypting the media content. The maximum DRM Service instances allowed would be equal to the number of HW key slots available in a HW platform.

3.2.5 TSB SERVICE

TSB Service encapsulates the nexus playback, record and transport modules to provide the time shifting functionality. Application provides the maximum number of media file segments that would be used for a TSB Service instance. TSB Service instances re-use the media/navigation file segments after reaching the max TSB buffer time limit to provide a constant TSB buffering time window. The maximum TSB service instances allowed would be equal to number of HW record and playback channels in a HW platform. TSB service doesn't control audio/video decoders and display HW. Application has to configure audio/video decoders and display through nexus APIs.

It provides the following functionalities -

1. Media probing of the buffered content to extract various audio/video stream parameters.
2. Start of recording callback to the application after the navigation entry for the first I frame is available.
3. Media probe complete callback to the application after all the audio/video parameters are extracted from the buffered content.
4. TSB conversion by copying the meta-data files. During TSB conversion, the media and navigation file segments are shared between permanent TSB converted recording and TSB buffering.
5. Encryption of TSB buffered and TSB converted recording through DRM Service.
6. TSB playback and trick modes.
7. Start of TSB playback and end of TSB playback callbacks to the application during TSB playback.
8. Injection of user provided PAT/PMT transport packets into TSB buffered recording or TSB converted recording.
9. PID remapping of audio/video PIDS.
10. Alarm callback to the application when app specific TSB playback time has reached during TSB playback.

3.2.6 RECORD SERVICE

Record Service encapsulates the nexus record and transport modules to provide linear or back ground recording functionality. There is no limit on how many media and navigation file segments are used for a linear recording since it depends on the duration of a recording. The maximum number of record service instances allowed would be equal to the number of HW record channels available in a HW platform.

It provides the following functionalities –

1. Media probing of the recording content to extract various audio/video stream parameters.
2. Start of recording callback to the application after the navigation entry for the first I frame is available.
3. Media probe complete callback to the application after all the audio/video stream parameters are extracted from the recording content.
4. Encryption of recording content through a DRM Service.
5. Injection of user provided PAT/PMT transport packets into recording content.
6. PID remapping of audio/video PIDs.
7. VOD Streaming from memory.
8. Acts as an output service for the trans-code service.

3.2.7 PLAYBACK SERVICE

Playback Service encapsulates the nexus playback and transport modules to provide playback functionality for segmented permanent recordings. The maximum number of playback service instances allowed would be equal to the number of HW playback channels in a HW platform. Playback service doesn't control audio/video decoders and display HW. Application has to configure audio/video decoders and display through nexus APIs.

It provides the following functionalities –

1. Playback of permanent recorded content
2. Playback of in-progress recording (either TSB converted or linear recording).
3. Start of playback and end of playback callbacks to the application during playback.
4. Trick mode support including seek operation.
5. Alarm callback to the application when app specified playback time has reached during playback.
6. Decryption of encrypted recording during playback.

3.2.8 HN STREAMING FILE INTERFACE

Media File interface acts as the HN Streaming file interface for the segmented recordings and provides basic file operations like open, close, read, write and seek for the segmented recording. Application specific HN modules can treat the segmented recordings as normal POSIX files using Media File interface file operations. There is no limit on how many media file interface instances can be created from HW side.

It provides the following functionalities –

1. Support for streaming from TSB recordings.
2. Support for streaming from permanent recordings.
3. Decryption of the encrypted TSB and permanent recordings through a DRM Service instance.
4. Server side trick mode support for streaming from TSB and permanent recordings.
5. Support to maintain continuity in PTS, DTS AND PCR during server side trick modes

3.2.9 TRANSCODE SERVICE

As shown in the FIGURE-10, trans-code service encapsulates nexus video window, trans-code display, video encoder, play pump PID channel, audio encoder and audio multiplexer.

The input of trans-code service can be a QAM input, HDMI input and playback service. As of now, only playback service input is supported. The output of trans-code service can be a record service or a TSB service and can be encrypted by associating the output service instance to a DRM service instance.

The trans-code output instance can be associated with a data injection service instance to insert the PAT/PMT transport packets.

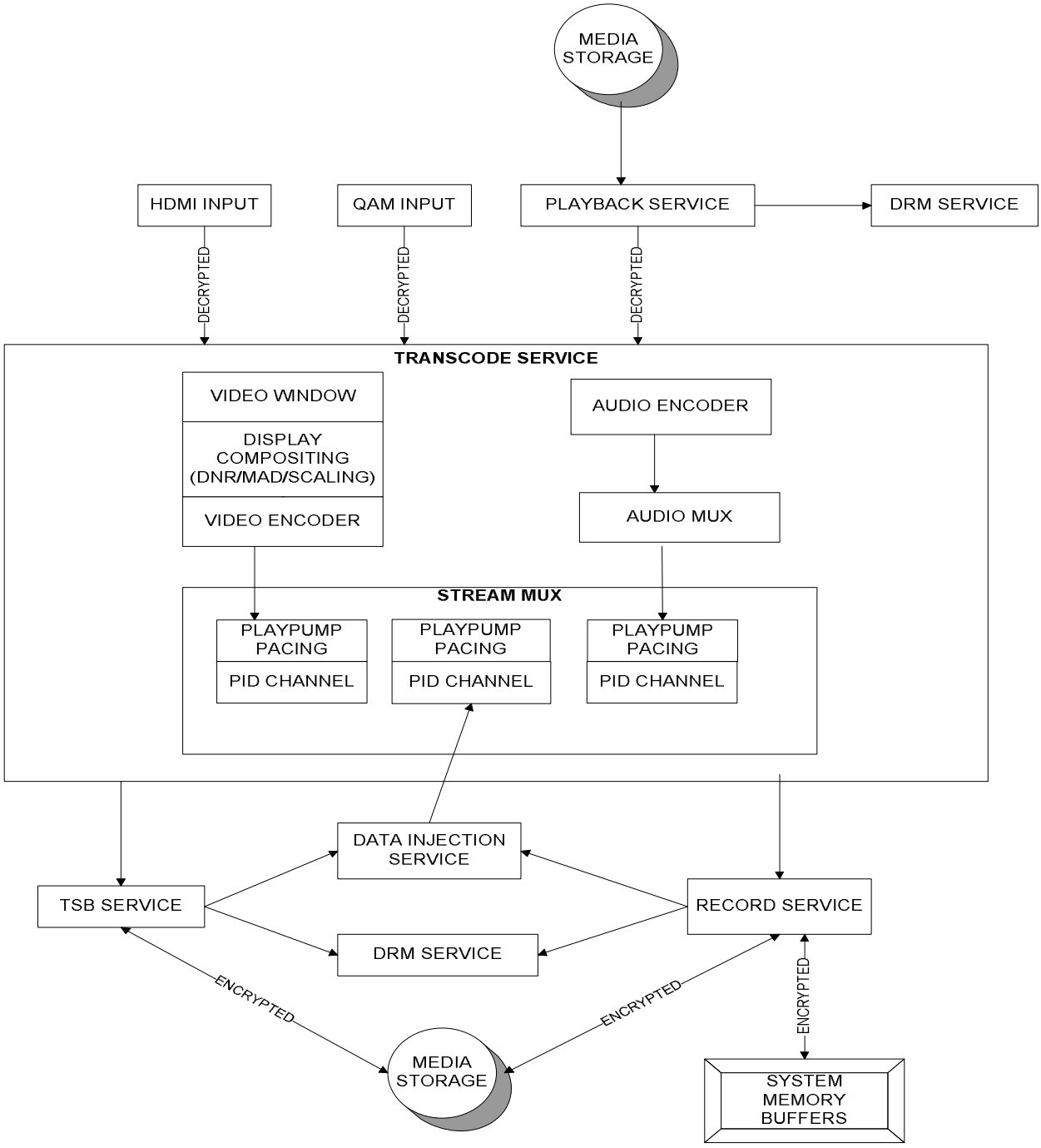
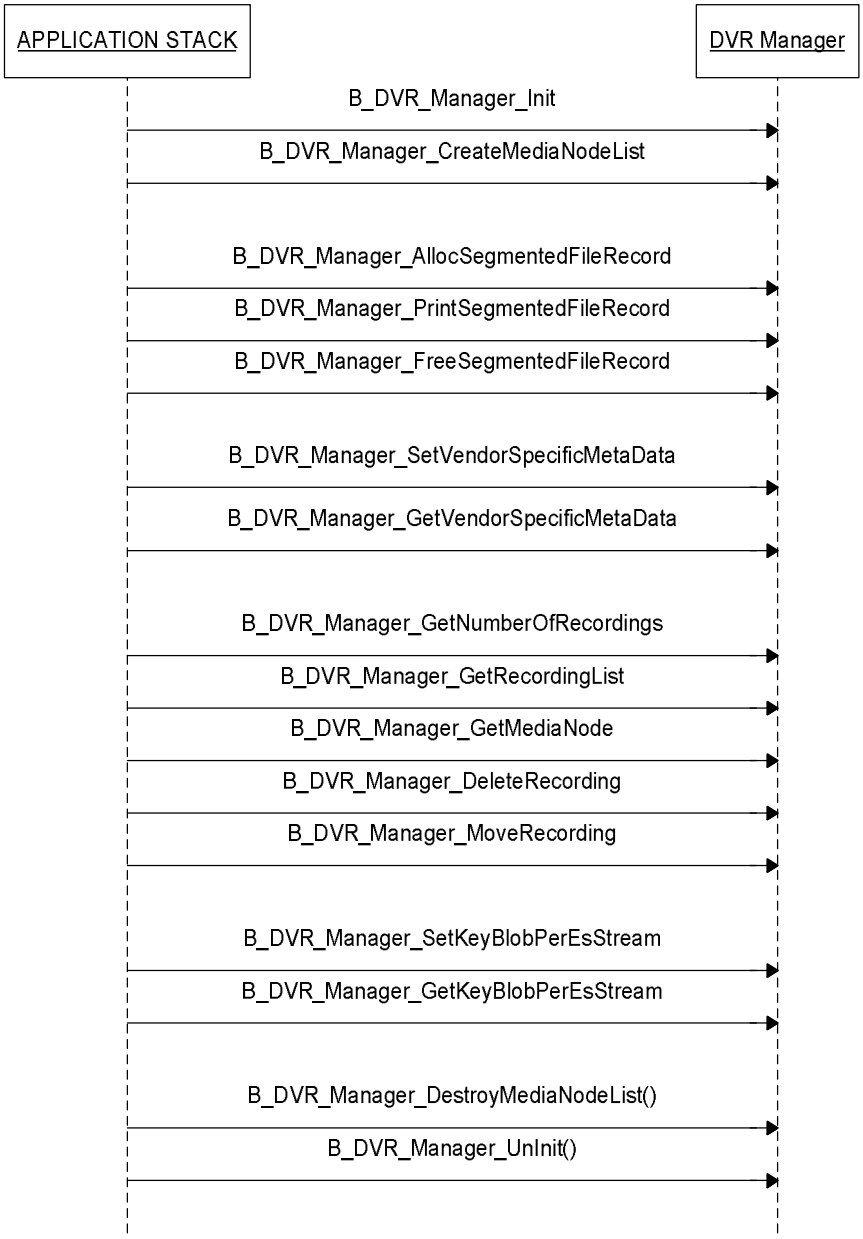


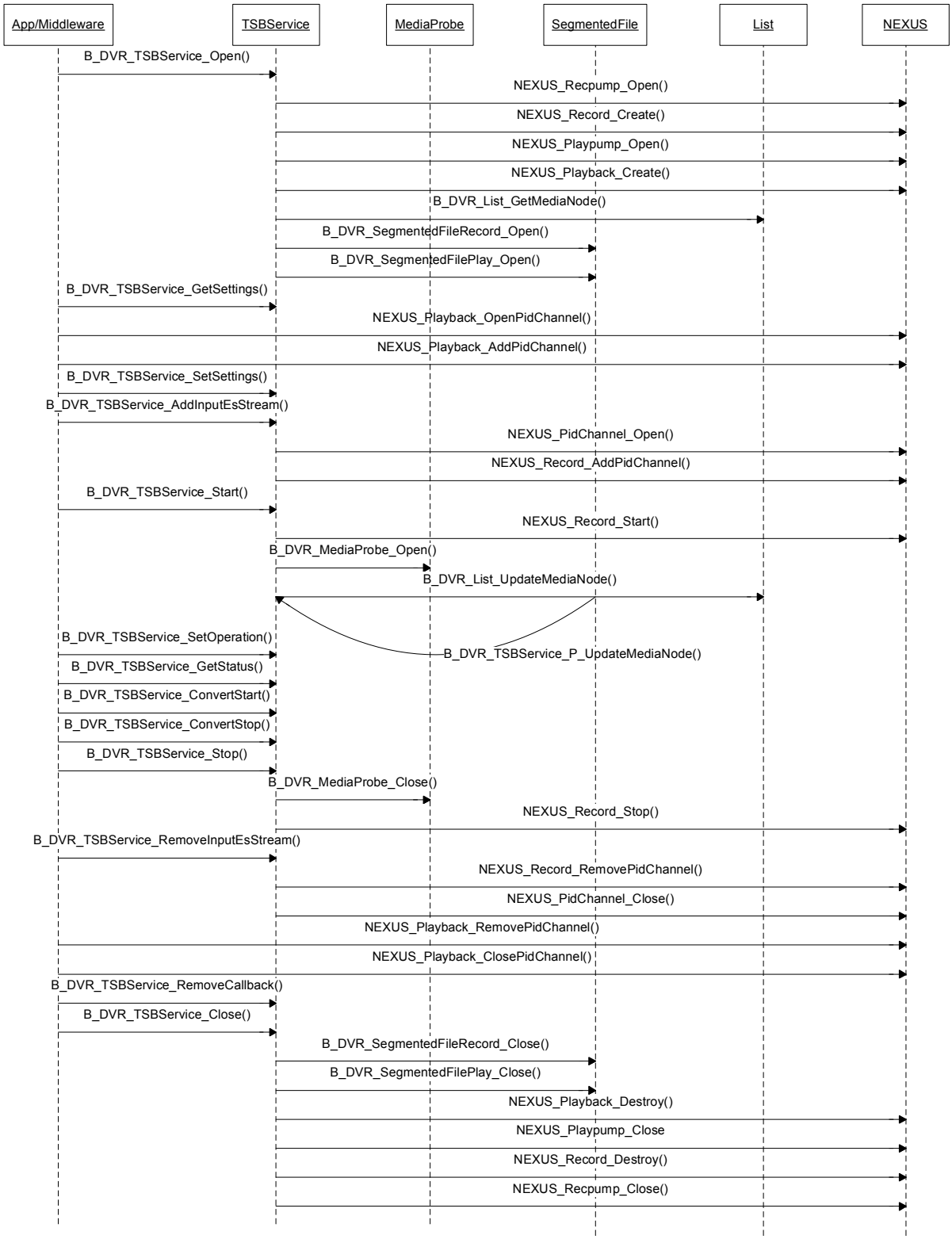
FIGURE-10. TRANSCODE SERVICE

4. DVR EXTENSION LIBRARY API SEQUENCE DIAGRAMS

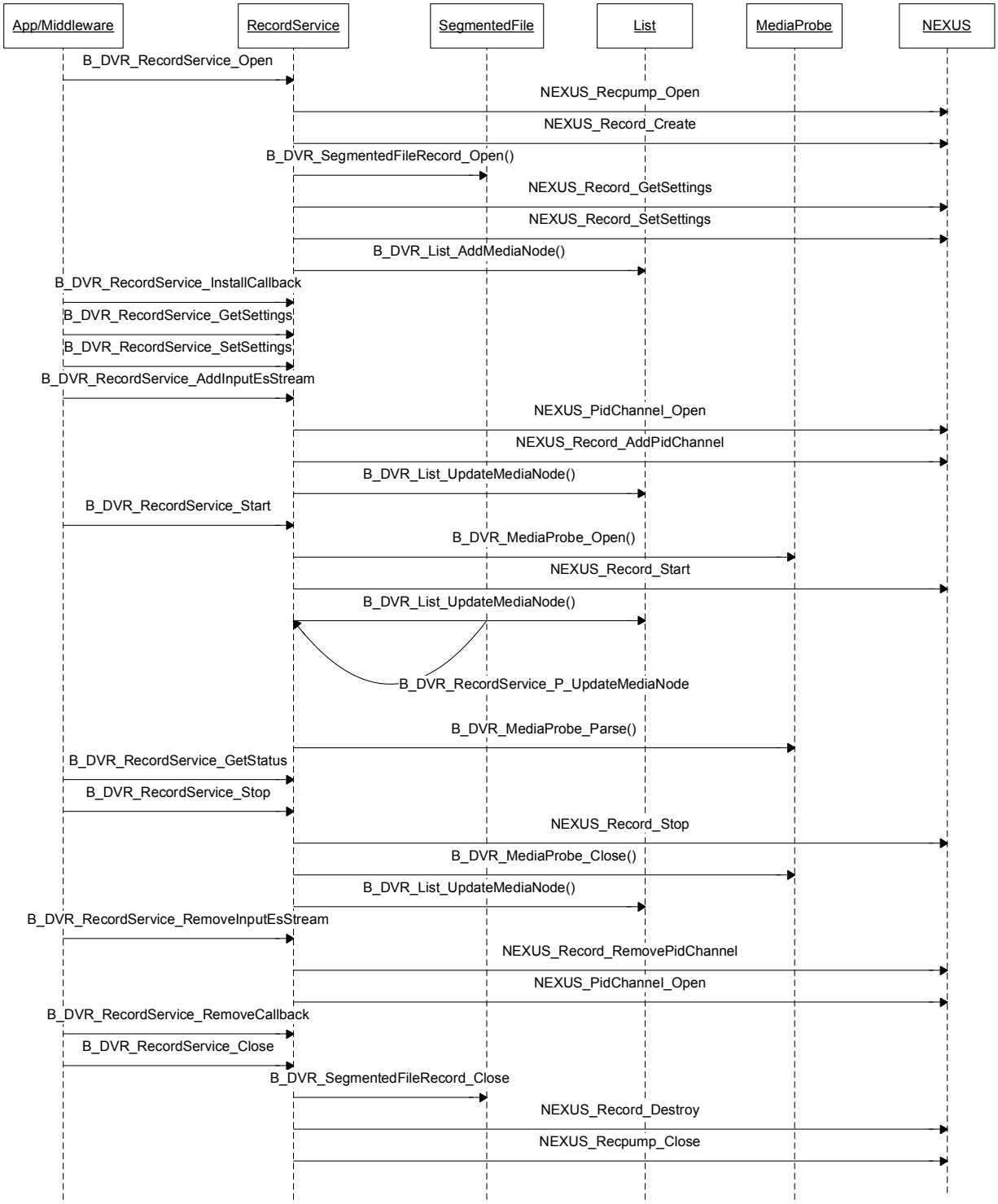
4.1 DVR MANAGER



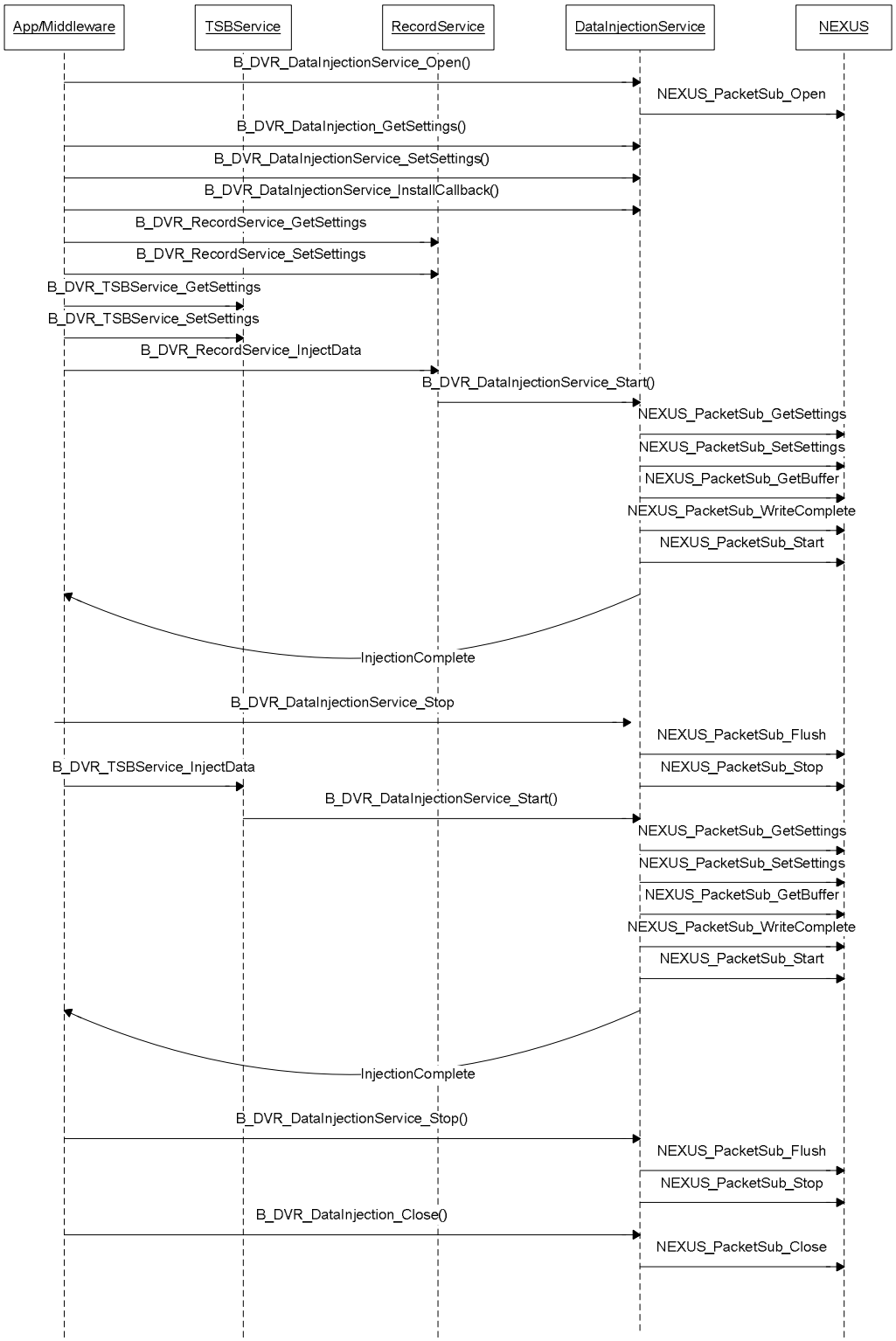
4.2 TSB SERVICE



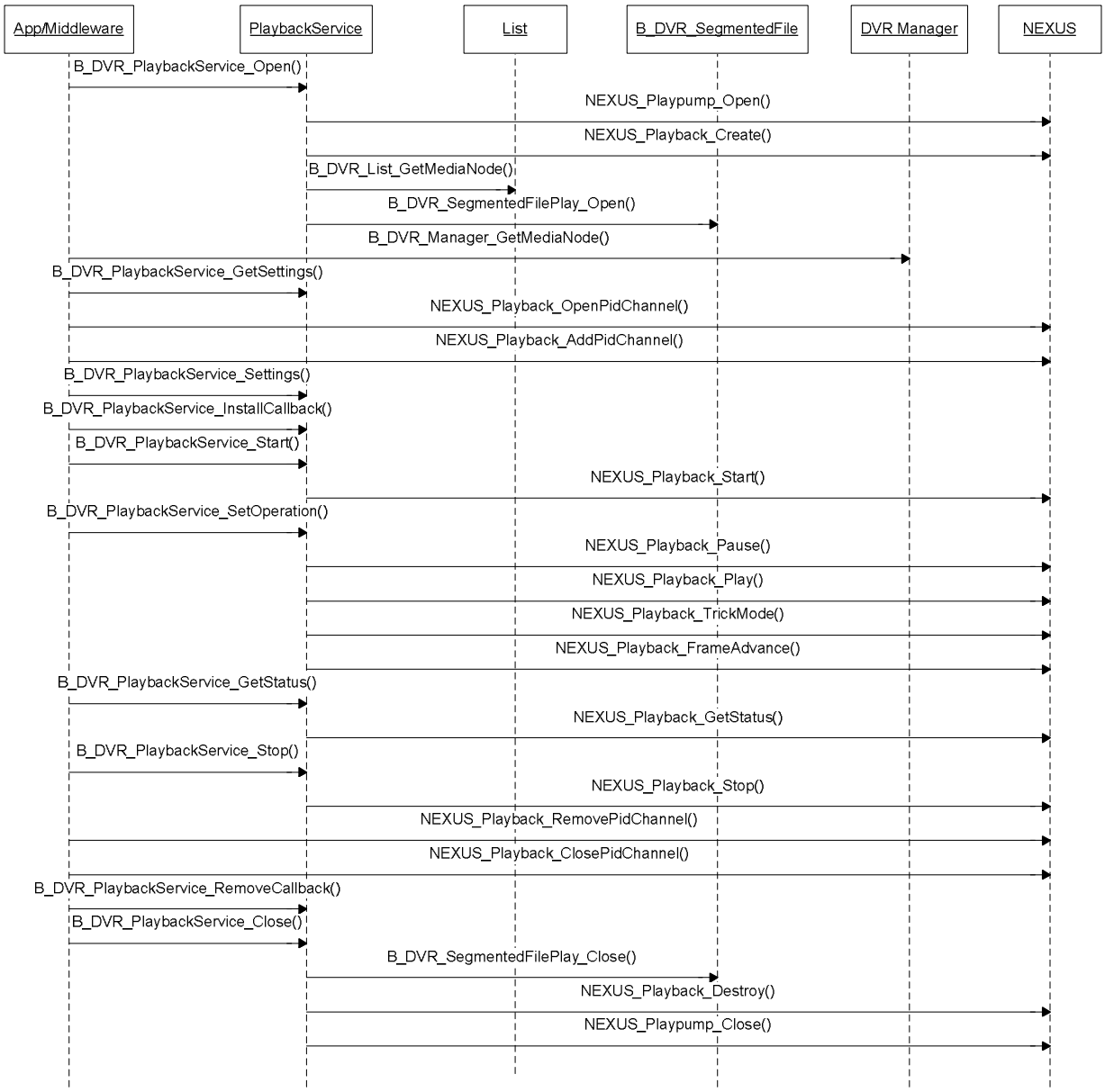
4.3 RECORD SERVICE



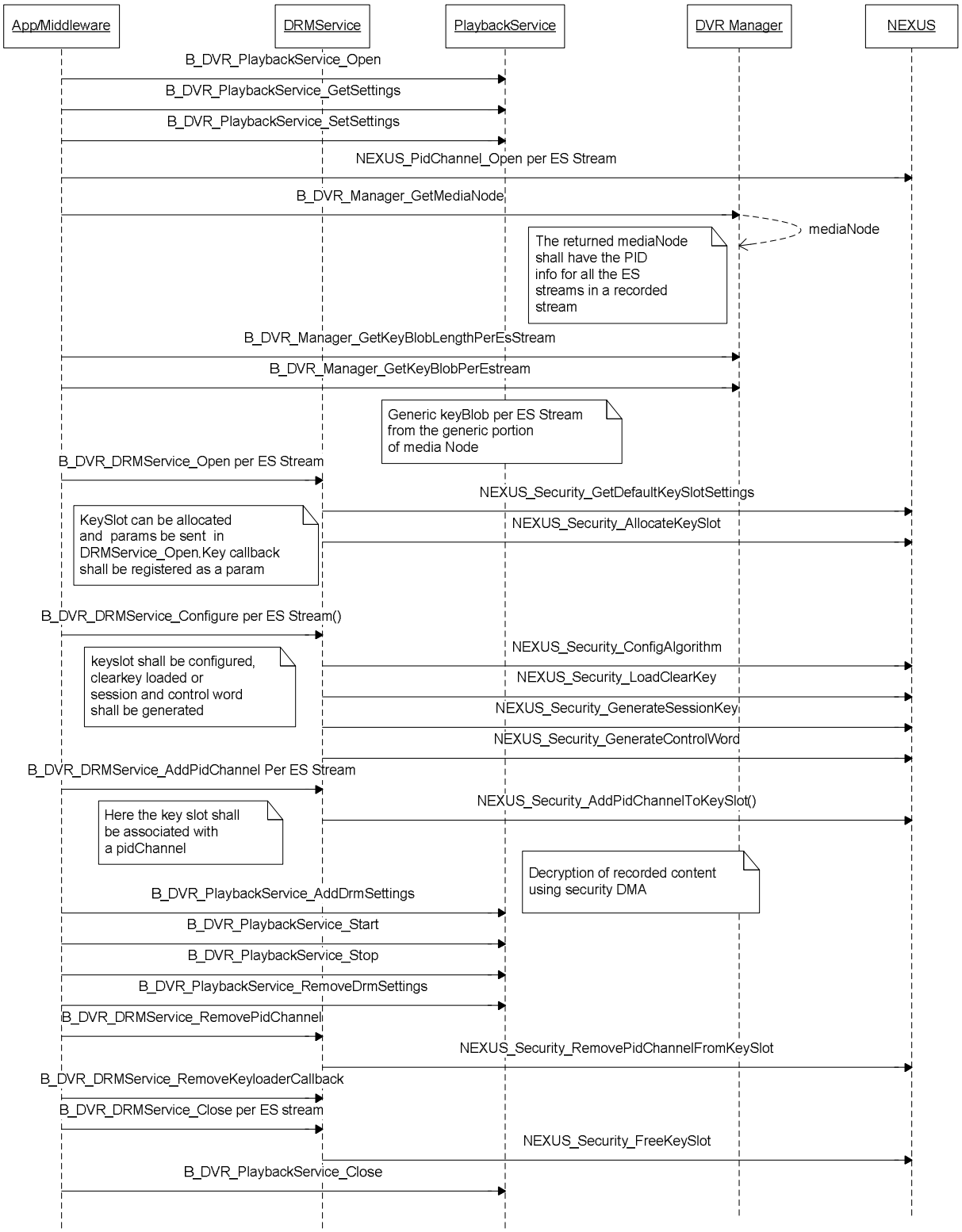
4.4 DATA INJECTION SERVICE



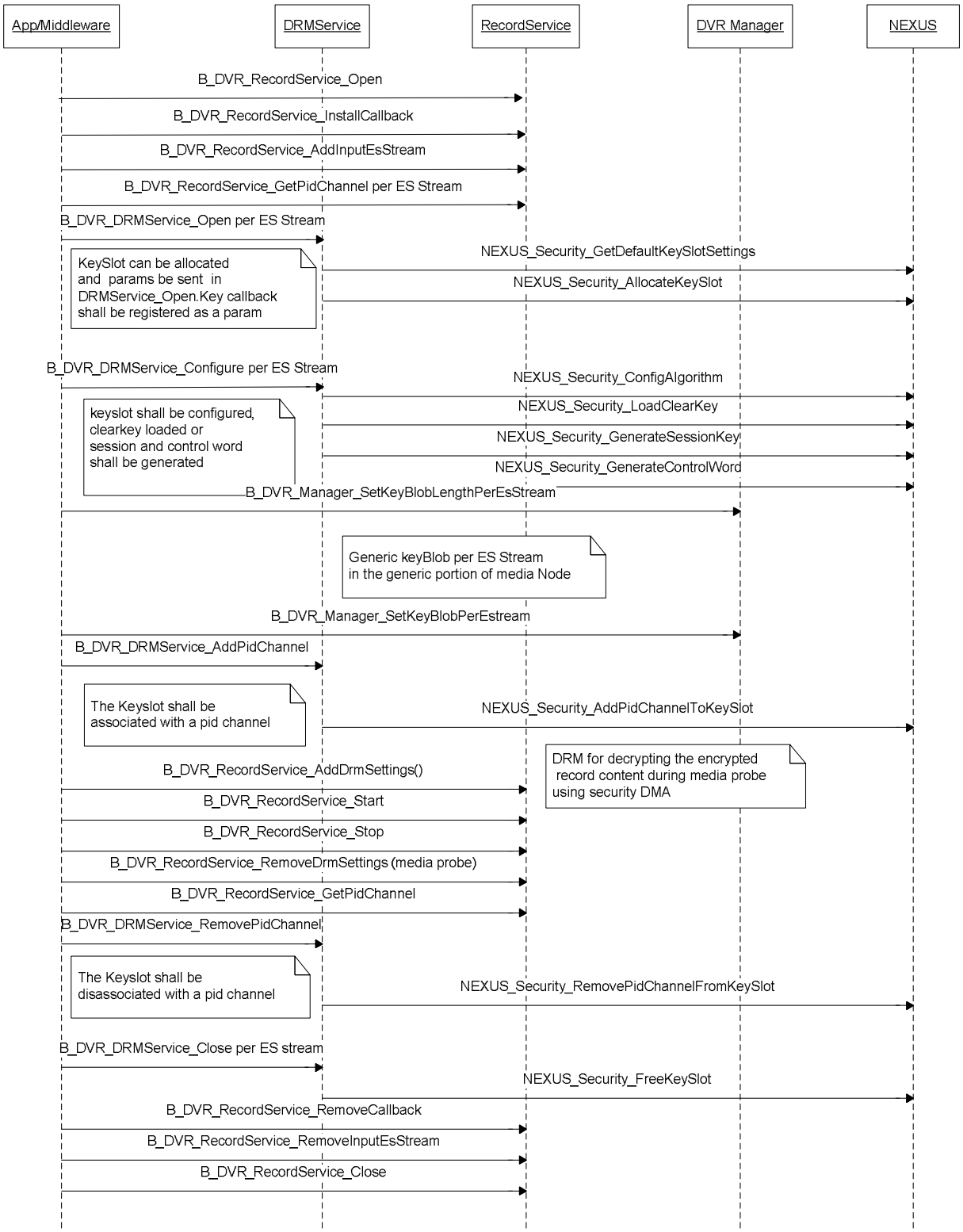
4.5 PLAYBACK SERVICE



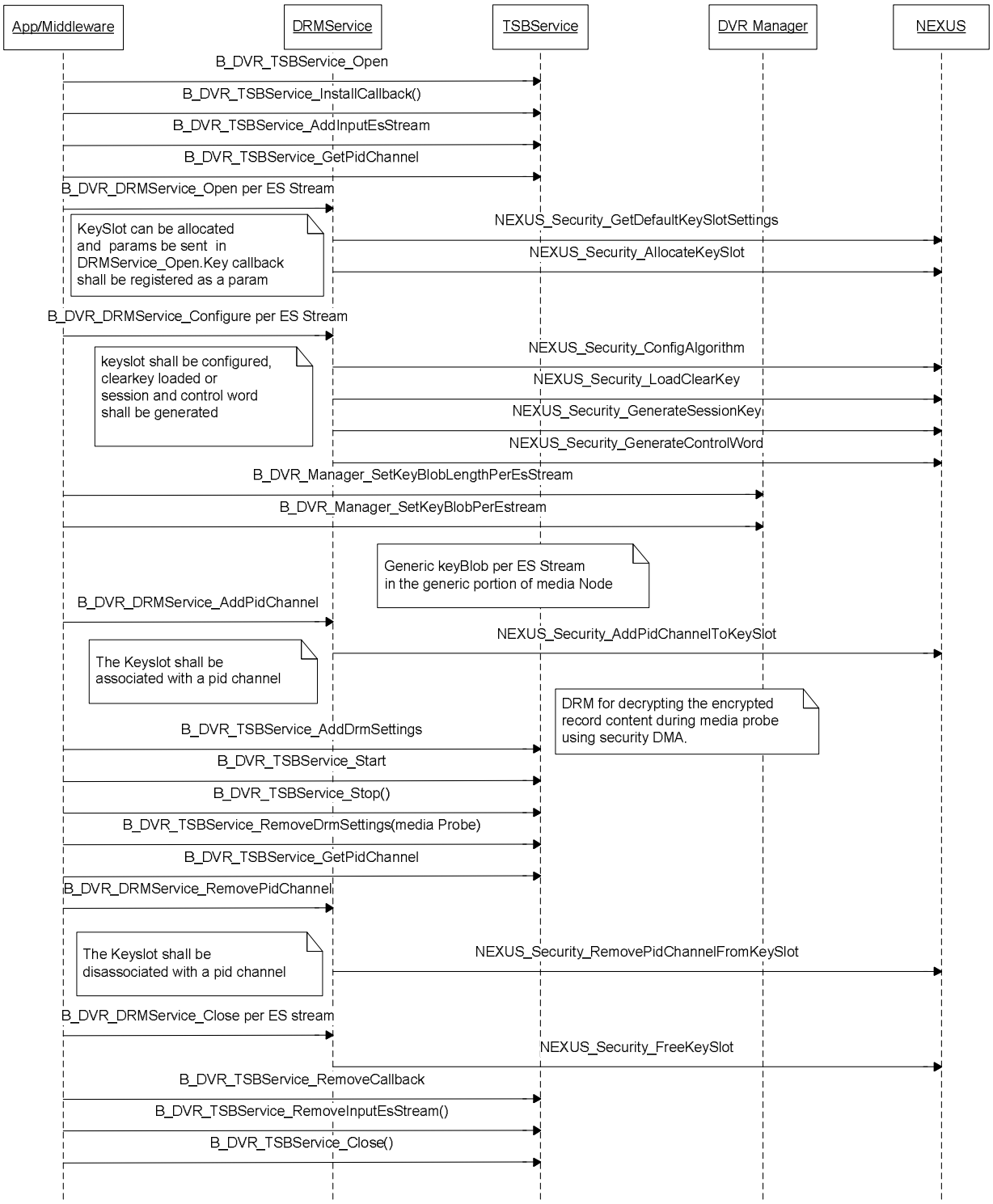
4.6 GENERIC DRM SERVICE IN PLAYBACK MODE



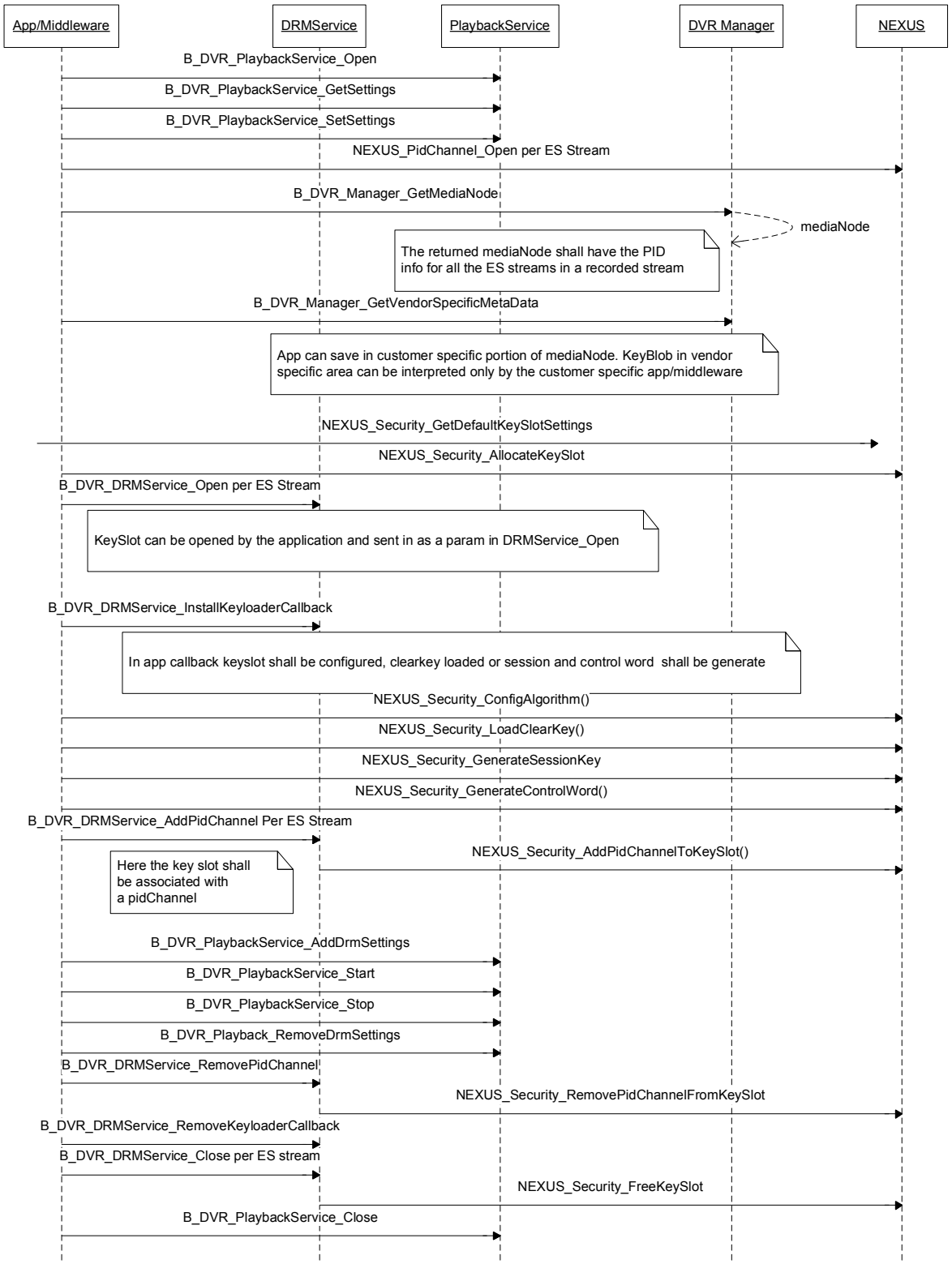
4.7 GENERIC DRM SERVICE IN RECORD MODE



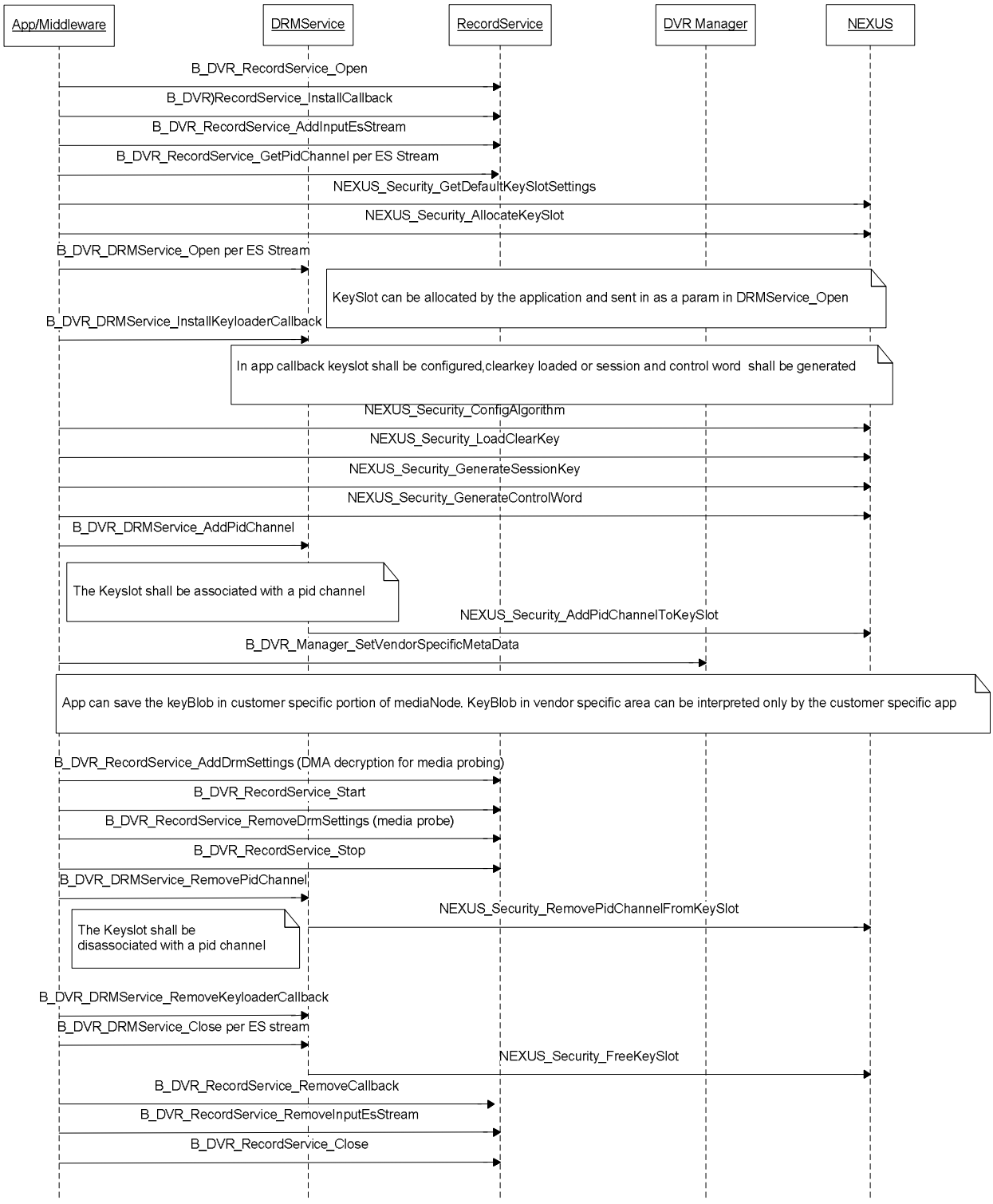
4.8 GENERIC DRM SERVICE IN TSB MODE



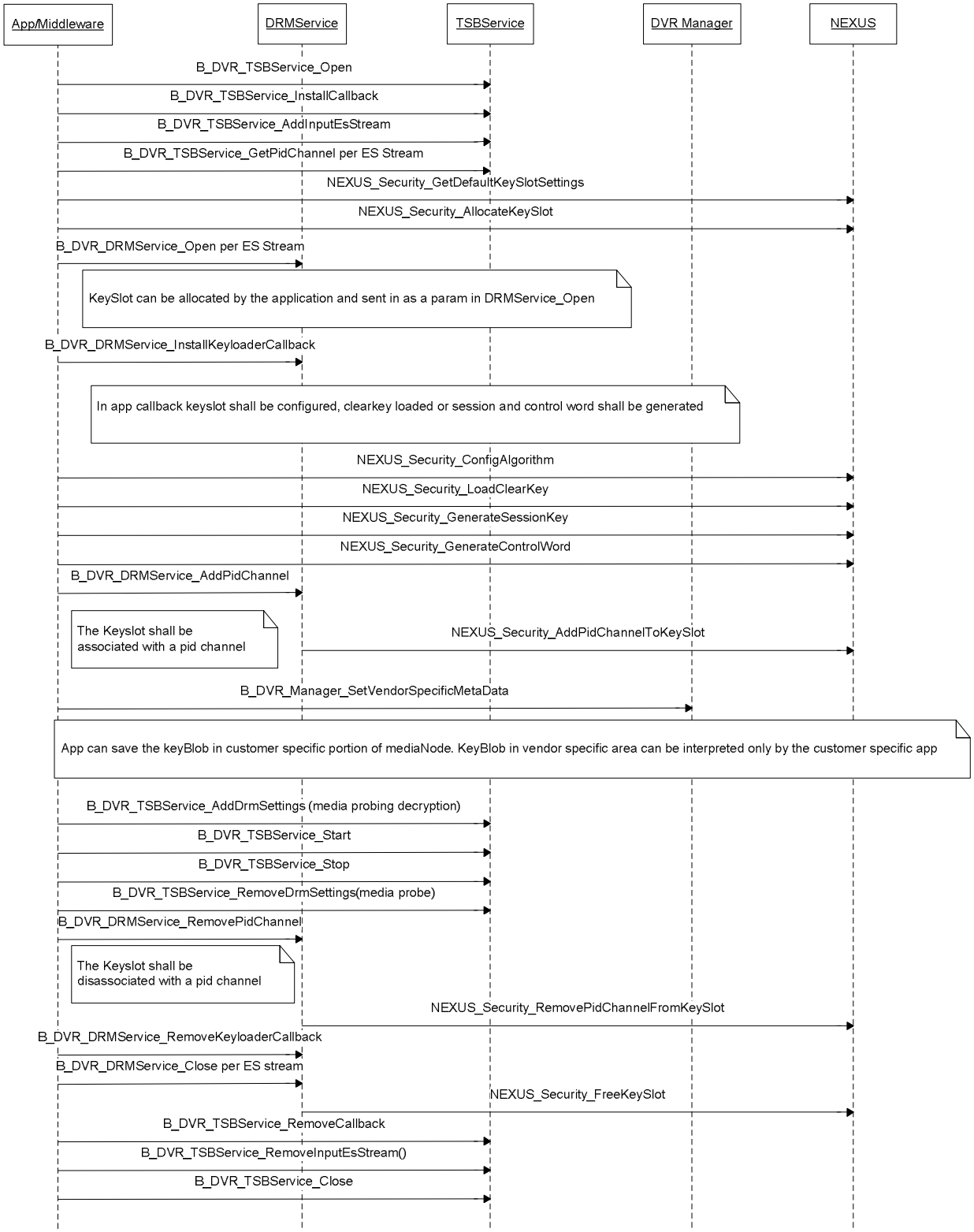
4.9 VENDOR SPECIFIC DRM SERVICE IN PLAYBACK MODE



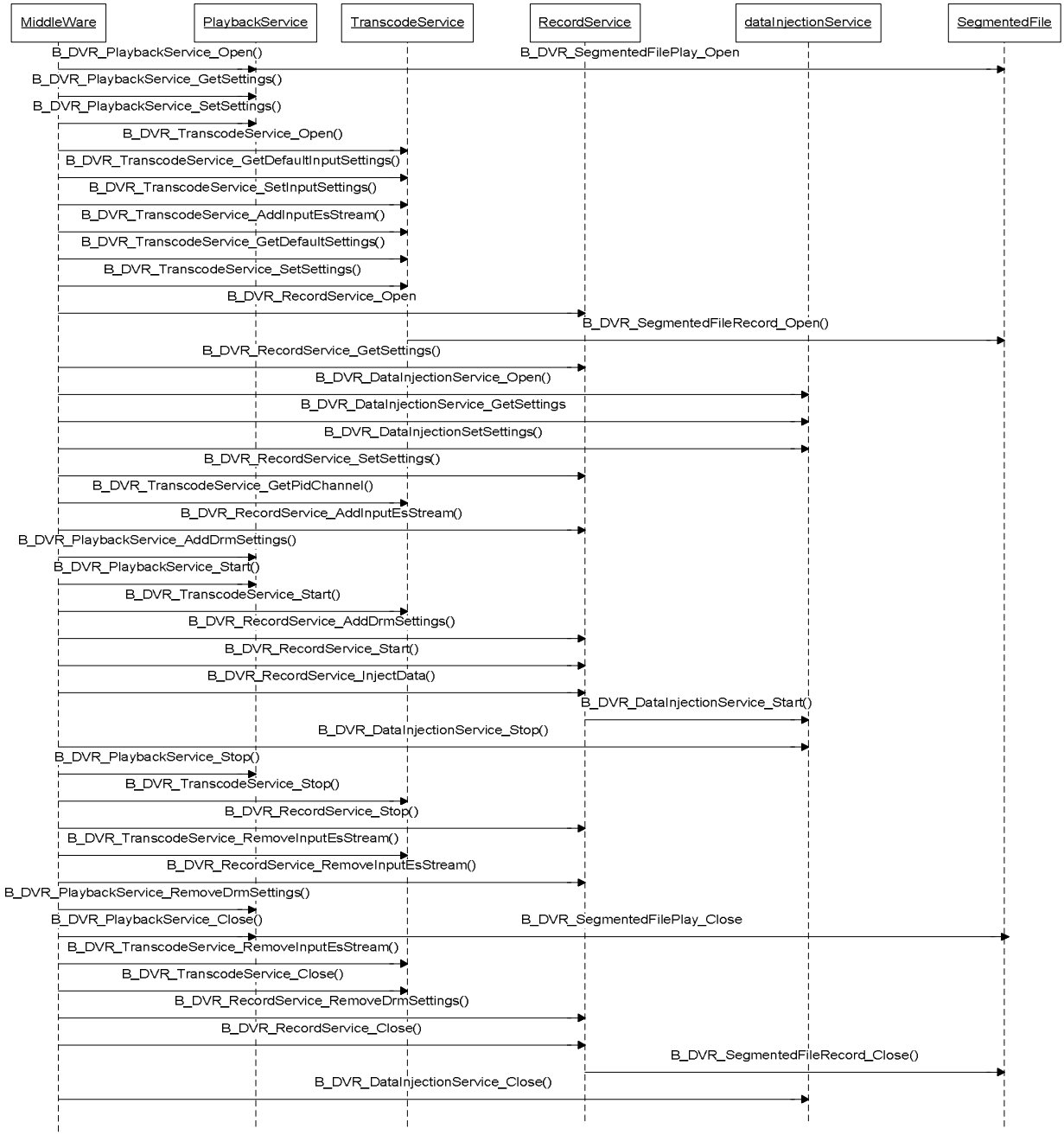
4.10 VENDOR SPECIFIC DRM SERVICE IN RECORD MODE



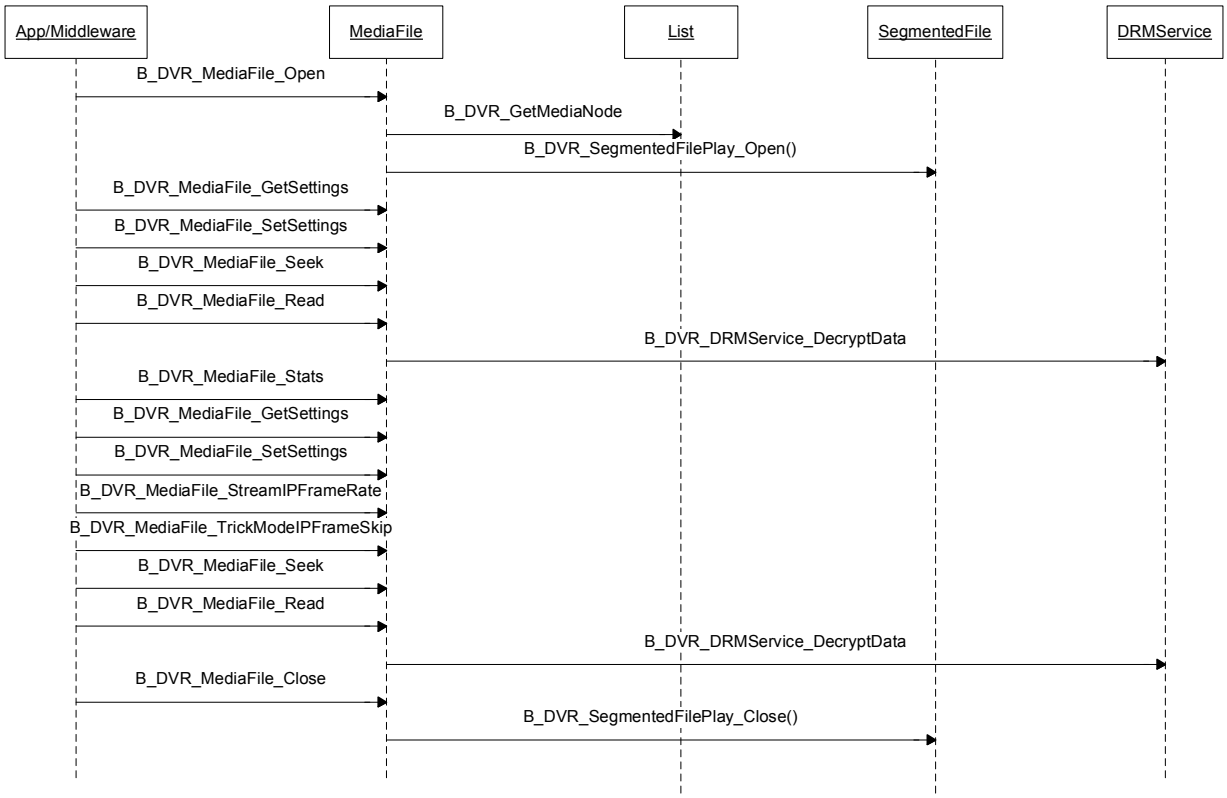
4.11 VENDOR SPECIFIC DRM SERVICE IN TSB MODE



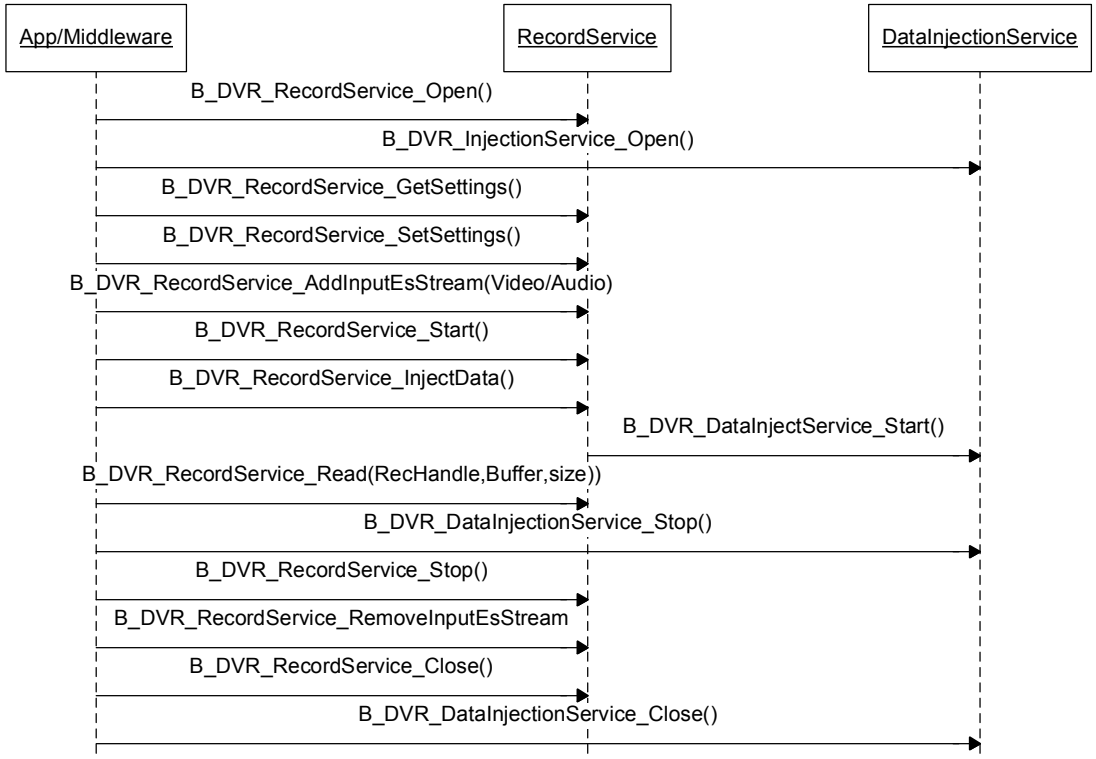
4.12 FILE TO FILE TRANSCODING



4.13 HN STREAMING FROM PERMANENT RECORDING



4.14 HN STREAMING FROM MEMORY



4.15 HN Streaming from TSB