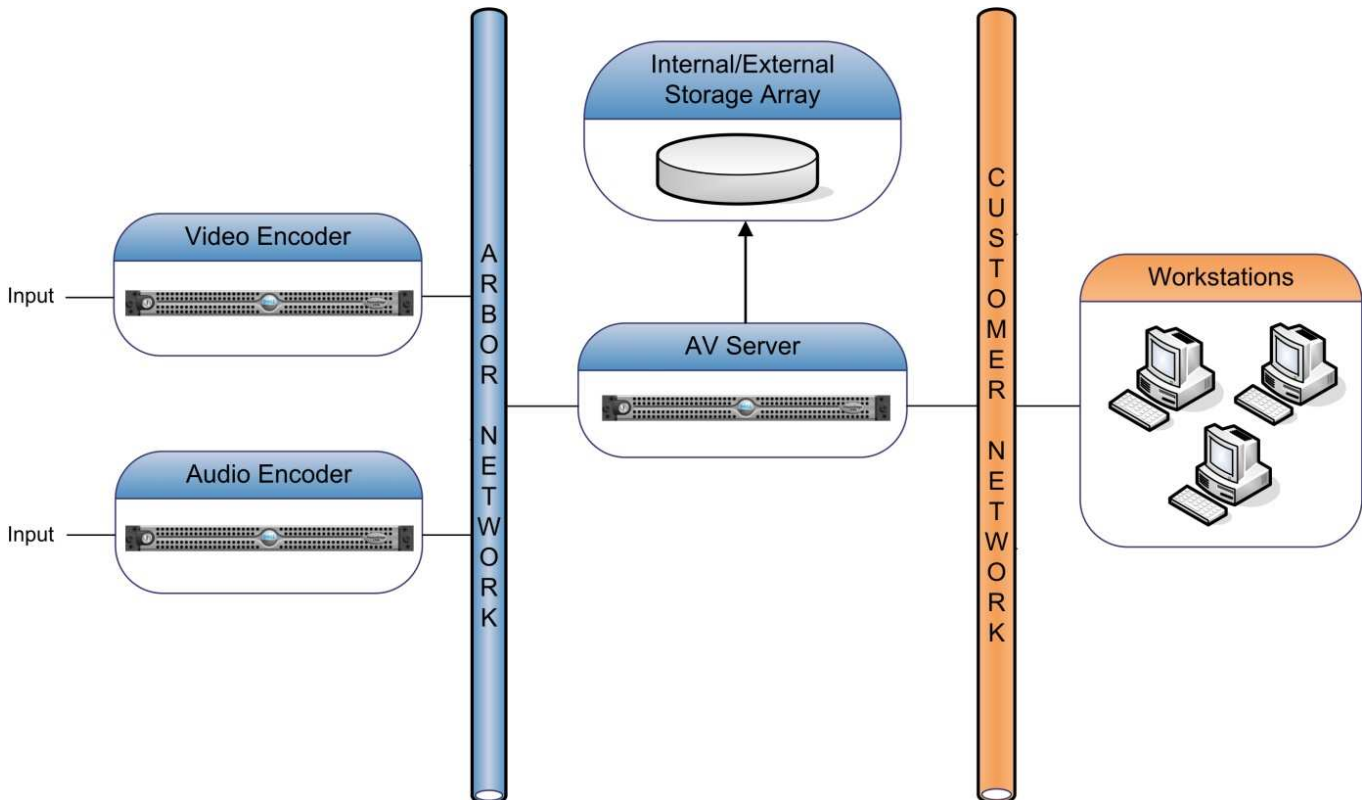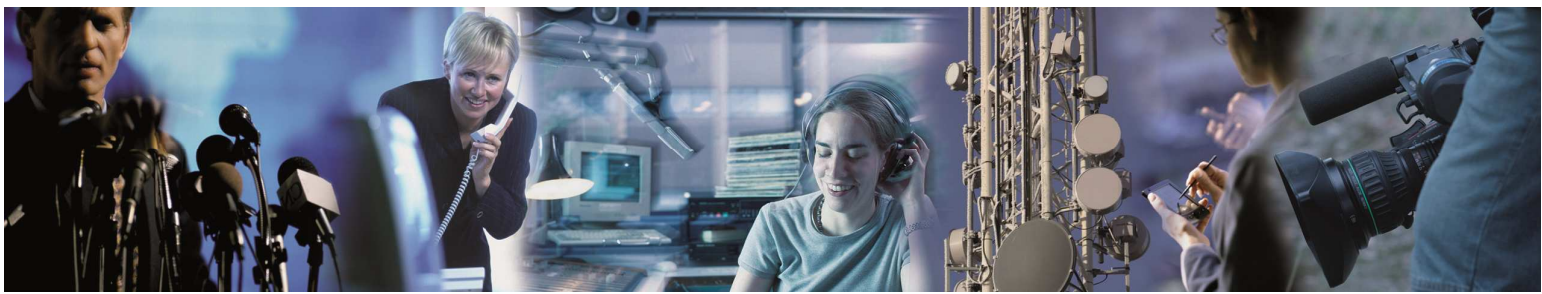# LogDepot Broadcast
Technical information

ARBOR MEDIA
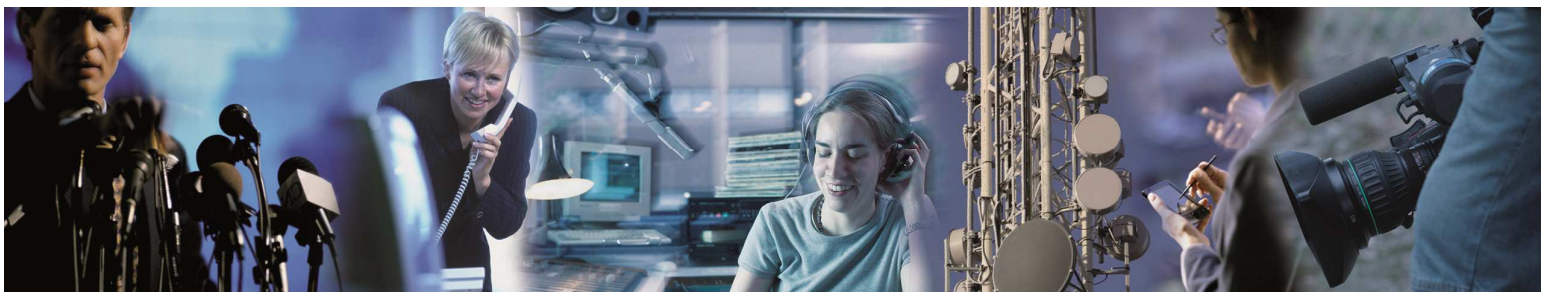
BROADCAST & CONFERENCE APPLICATIONS

The Concept

The LogDepot architecture is powerful, yet simple in design. It was designed to be robust and reliable, but also scalable, flexible and efficient.

A standard LogDepot system is shown in the picture below

The audio/video backends receive the input signals and record the signal on their local disk, while encoding or transcoding to the required formats. The system is capable of handling a number of input signals on one backend server PC: varying from a few to dozens. And there is virtually no limit to the number of backend servers, so scaling up the system is as simple as adding another backend server PC.

The fetcher process, residing on the AV server, continuously fetches small parts of the AV backends and stores the media onto the central storage.

The webclient, also residing on the AV server, provides the GUI (Graphical User Interface) for the users on the customer network. The pictures below show an audio/video and audio-only version of the web based LogDepot GUI. On the next page the LogDepot Audio web client is explained and shown.

ARBOR MEDIA
BROADCAST & CONFERENCE APPLICATIONS

Recording Audio with the Audio Logger

The audio logger can receive audio input signals from a variety of inputs, including Balanced analogue, AES/EBU, ADAT, MADI, ASI, TS-over-IP, DVB-S, DVB-C, DVB-T.

The audio backend software module, running on the audio logger, will encode or transcode the incoming audio inputs to a relatively low-bit rate Proxy format, suitable for browsing and optionally a high quality signal to be used for archiving and as source for other export processes. The high quality format can be original format (Linear PCM or the original TS/ DVB format).

The LogDepot is able to record **multiple streams** in multiple formats from one physical input.

Audio backend

The audio backend stores the material in a cyclic buffer on the local disk. The size of the local storage is typically enough to overcome at least a couple of days of network and/or server failure (see also: Reliability and redundancy).

The audio backend user interface shows the status and the modulation level on all audio inputs.

Supported output formats

- Windows Media Format
- MPEG1, Layer II
- Linear PCM
- AAC (MPEG4)

*These are **native** storage formats; conversion to many other formats is possible with the export process.*

ARBOR MEDIA
BROADCAST & CONFERENCE APPLICATIONS

Recording from traditional audio inputs

When recording traditional audio inputs, such as balanced analogue, AES/EBU, ADAT or MADI, the audio backend runs on a standard server, equipped with an ADAT or MADI interface card. Break-out boxes are used to convert balanced analogue and AES/EBU signals to ADAT or MADI, and thus provide the interface to audio backend server PC. One backend server can handle up to 64 audio inputs.
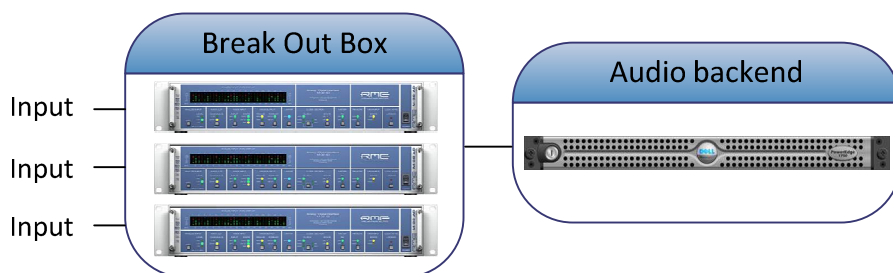

Recording Radio programs from DVB Transport Streams

When recording Radio programs, the source is more and more likely to be digital: from Satellite, Digital Cable or Digital Terrestrial networks. The audio backend can handle these digital DVB Transport Streams in their native format.

A special AV Gateway is used to receive (and if necessary decrypt) the Transport Streams and transform to a TS-over-IP stream. The audio backend can receive the TS-over-IP stream on its regular network interface (no additional hardware interface cards required) and record the audio programs.

The digital input signal can be stored in its original (native) format or can be transcoded to a Proxy stream and/or to a High Quality stream as mentioned before.

The input stream may be in SPTS (Single Program Transport Stream) or in MPTS (Multi Program Transport Stream). In the latter case the audio backend will demultiplex the TS and store each recorded channel as a single stream.
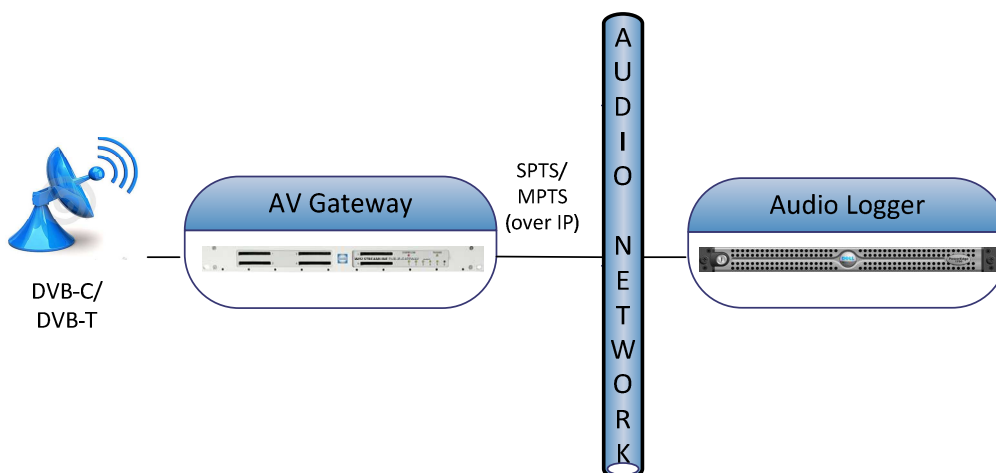
Recording Video with the Video Logger

The Video Logger will receive the video input signals from a variety of inputs, such as Composite Video, Analogue HF, SDI, ASI, TS-over-IP, DVB-S, DVB-C, DVB-T.
The video backend module, running on the Video Llogger will encode or transcode the incoming video inputs to a relatively low-bit rate Proxy format, suitable for browsing and optionally a high quality signal to be used for archiving and as source for other export processes. The high quality format can be original format (the original TS/ DVB format).

The LogDepot is able to record **multiple streams** in multiple formats from one physical input.

The video backend stores the material in a cyclic buffer on the local disk. The size of the local storage is typically enough to overcome at least a couple of days of network and/or server failure (see also: Reliability and redundancy).
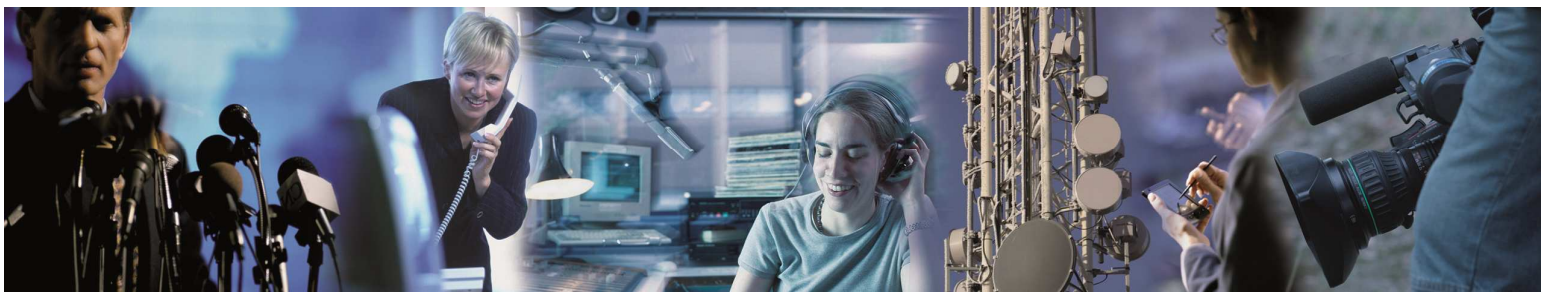
The video backend user interface shows the status and preview of all video inputs

Supported output formats
- Windows Media Format
- MPEG2
- H.264/AVC (MPEG4)

*These are **native** storage formats; conversion to many other formats is possible with the export process.*

Recording traditional video inputs

The Video Backend runs on a standard (server-) PC, equipped with traditional Video Capture cards with HF analogue, Composite Video or SDI interface. The number of video inputs on one Video Backend server is usually limited by the hardware interface capabilities.
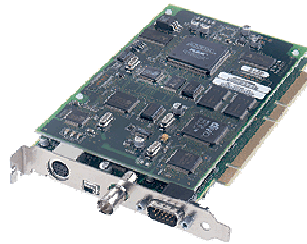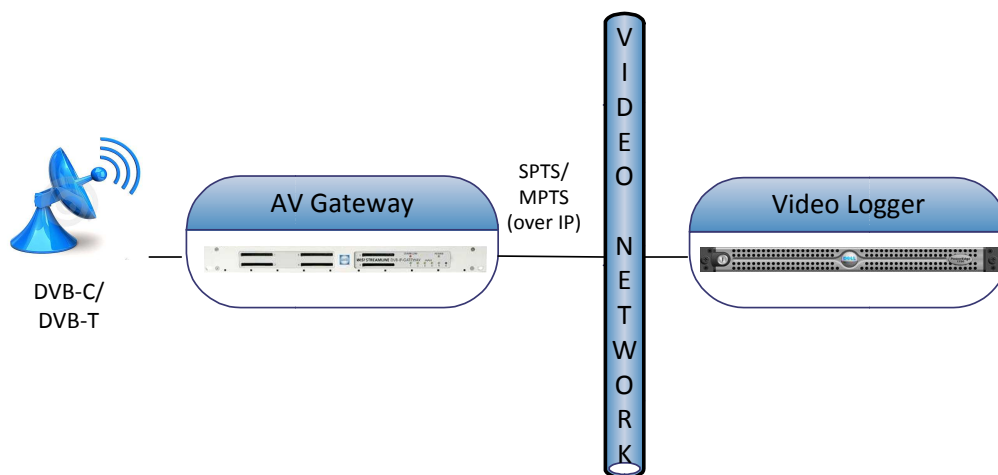

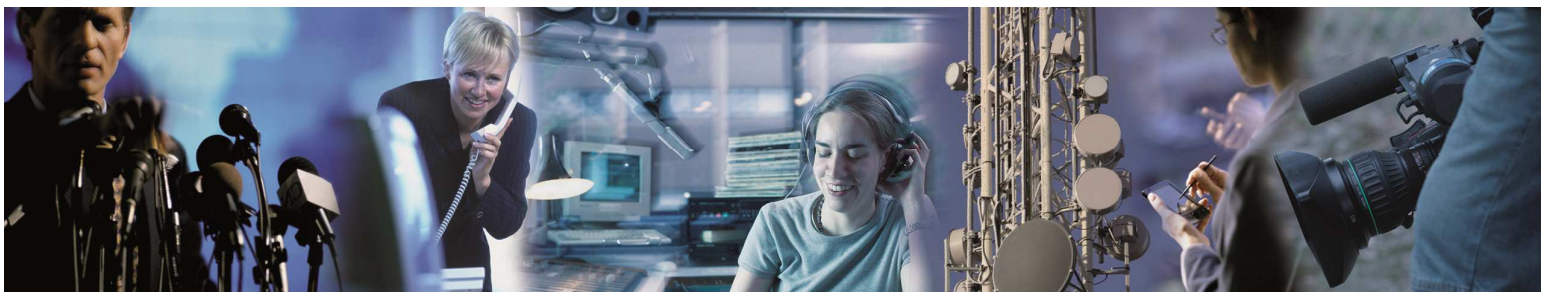## Recording TV programs from DVB Transport Streams

When recording TV programs, the source is more and more likely to be digital: from Satellite, Digital Cable or Digital Terrestrial networks. The Video Backend can handle these digital DVB Transport Streams in their native format.

A special AV Gateway is used to receive (and if necessary decrypt) the Transport Streams and transform to a TS-over-IP stream. The video backend can receive the TS-over-IP stream on its regular network interface (no additional hardware interface cards required) and record the these programs.

The digital input signal can be stored in its original (native) format or can be transcoded to a Proxy stream and/or to a High Quality stream as mentioned before.

The input stream may be in SPTS (Single Program Transport Stream) or in MPTS (Multi Program Transport Stream) format. In the latter case the backend will demultiplex the TS and store each recorded channel as a single stream.

Multilingual Recording

Multilingual TV programs can be recorded as one stream, containing all the available language tracks. In case of multilingual environments, the user obtains the possibility to choose the preferred language track in the user interface.

When recording TV programs, the source is more and more likely to be digital: from Satellite, Digital Cable or Digital Terrestrial networks. The video backend can handle these digital DVB Transport Streams in their native format.

The fetcher process

The fetcher module, running on the central AV server, is responsible for the storage of the media files onto the central storage system. The server can use internal Hard disks, DAS (Direct Attached Storage) via SAS or Fibre channel adapters, NAS (Network Attached Storage) or SAN (Storage Area Network) to archive the recorded media files.
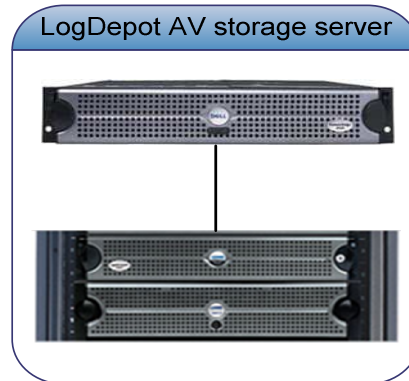
AV Server with Direct Attached Storage Array

The fetcher module retrieves all the stored files in the cyclic buffers from each AV Logger. This process takes place "on the fly", so that the delay is minimized. Media files can typically be accessed by client workstations over the customer network within 15s for audio and 30s to 60s for video. The central server with storage keeps the media files for a limited amount of time, varying from a few days to a few years and automatically takes care of the cleaning process. The size of the storage duration may be different for the different filetypes: it many cases, customers choose to keep the low bitrate Proxy files longer than the High Quality files.

If parts of the recordings must be archived permanently, the Export Server is used to execute this process (automatically, if possible).

LogDepot supports the use of multiple servers, in which cases each server will collect the media recordings in a continuous process.
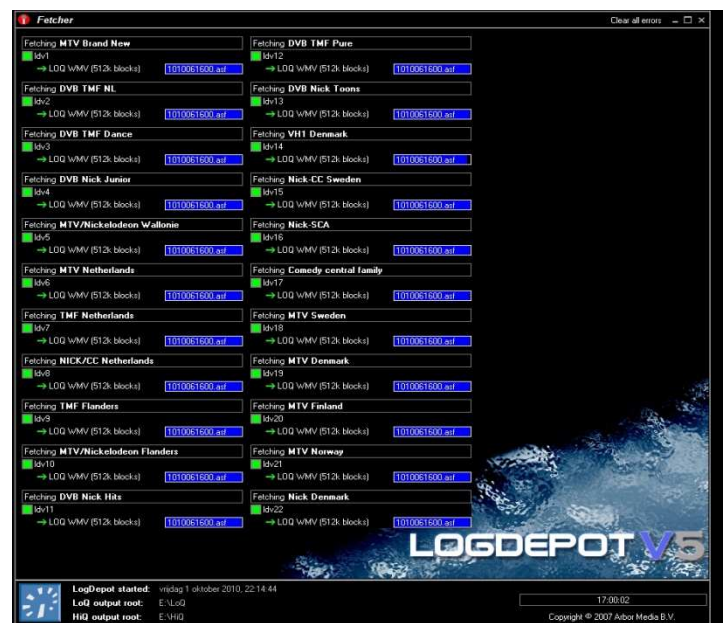
Reliability and redundancy

The most important feature of any Audio/Video recording system is the reliability of the recording itself: Material may not be lost due to any hardware failure. This fact ánd the knowledge that any hardware part can fail, were important factors for the design of the LogDepot architecture.

The Logger applications (Audio and Video backend) have been designed só that there only task is capturing and digitizing the incoming AV signals, convert them to the required, native media formats and store these on their local hard disk. The backends are not aware of the rest of the system, neither the AV server, nor the network, nor the users. This is the only way to warrant that network and/or server failures will not affect the basic recording process: Even when the whole network stops working, the loggers will continue to record.
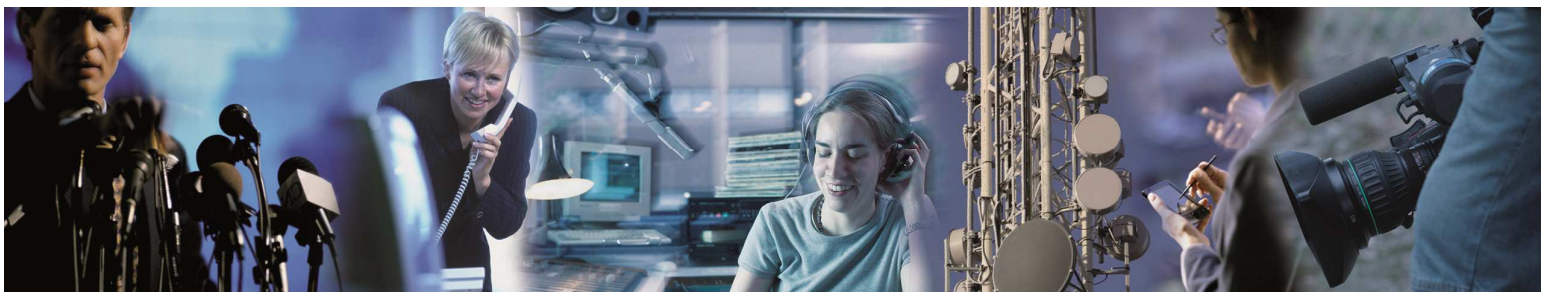
The Fetcher process on the AV server is responsible for the gathering of recorded material onto the central storage. The Fetcher "knows" what it needs (e.g. Television Channel XYZ) and where to get it (on Logger #3 or on Logger #4). It will attempt to fetch the media files from Logger #3, and if this fails try to fetch from Logger #4. If both (or better: all) attempts fail, it will retry after a while and continue to do so, until the attempt succeeds.

This means that LogDepot can be designed completely redundant, when used in mission critical environments, which means that all encoders are mirrored. If an encoder fails, the server will automatically detect the problem, and will switch over to the mirrored counterpart. Failure of an encoder will not lead to any loss of video material.

The encoders have local buffers to keep a few days (typically) of media recordings. This can overcome a few days of network or server failure.

Apart from mirroring the Loggers, it is also possible to mirror the LogDepot server and storage (typically in another geographical location).
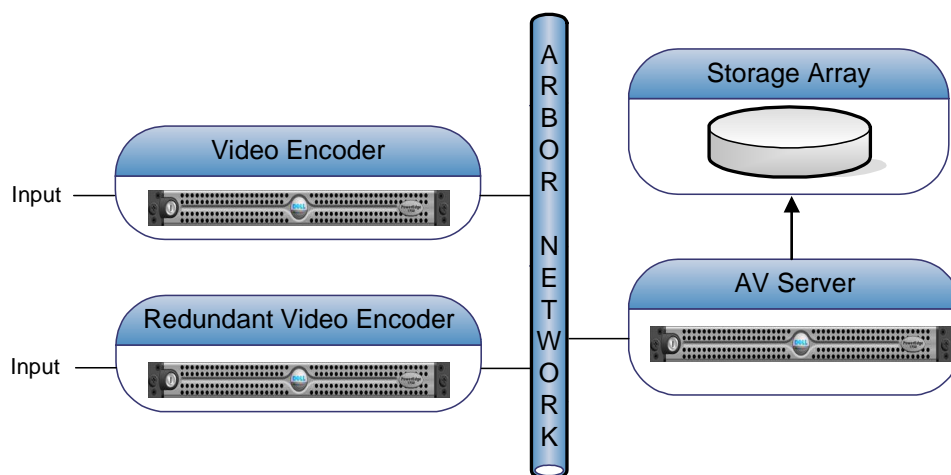
Metadata handling

The LogDepot architecture involves an unique, generic metadata handling mechanism. It can handle an user-definable metadata structure, allowing each customer to define metadata fields according to his requirements.

The concept for handling the metadata is similar to the handling of the media files. Various metadata backend modules can read metadata from a particular source, convert the structure to LogDepot-internal generic format and store this locally. The metadata fetcher, running on the AV Server, will fetch metadata from an unlimited number of metadata backends and store it in a central metadata database.

The following metadata modules are (among others!) available for broadcast applications:

DVB EPG   Read the metadata available in the DVB Stream. The availability depends on the provider, but it typically includes Program name, Sub title, Description and Category, but may also include other fields (see picture below)

PDC   Program Delivery Control; reads the Program name and category from the VBI information, typically available in analogue TV signals

RDS   Radio Data System; receives information such as Radiotext, Traffic Information, Music/Speech from an RDS source. The backend can read the RDS from a DVB stream or directly from a UECP stream

XML   Read metadata from any XML or Text file, including "As Run" lists. A built-in XSLT (XML Converter) allows a customizable conversion from the source file to the LogDepot internal presentation

Custom   Customized metadata backends can be created to capture metadata from special sources, including customer databases, internet sites or others.

The LogDepot can handle all sorts of metadata, there is no need to stick to a dedicated scheme, and – more important – metadata schemes may vary per video channel.

Metadata	
Program Title	The Inspector Lynley Mysteries
Program Sub Title	In the Blink of an Eye
Description	Britse detectiveserie. De moord op een paparazzo-fotograaf leidt Lynley en Havers naar zijn
Presenter	-
Director	-
Actor	Nathaniel Parker, Sharon Small, Danny Webb, Indira Varma
Release date	-
Category	Drama, Misdaad, Serie
Language	-
Duration	90 (minutes)

Export & Processing

The Export Server and Process Engine allow the customer to set up export profiles according to his own requirements. The configuration may be complex, as it requires lots of detailed information, but the result of a correct configuration, is an extremely powerful and easy-to-use export mechanism.

The export process will cut out the user-defined part(s) of (High Quality) video file, assembling them into one file if required and hand this file over to the processing engine, accompanied by an XML file containing all the relevant metadata.

The processing engine is able to convert audio, video and metadata and deliver the result to the target endpoint. Typically, the endpoint is folder on a server, but there are also more specialized endpoint available, such as the CD/DVD Robot Server, which will automatically author the audio/video files, burn the CD/DVD and print the label.

Processing modules include - among others - the possibility to:

- Convert Audio
- Convert Video
- Extract & embed metadata
- Deliver to fileservers
- Deliver to archiving systems
- Burn CD/DVD

To support additional workflows, an administrator can design a special "export form", requiring the user to enter specific information for this export process. This way, the administrator can require the user to enter a category, an ID, a description or anything else, before the process is started. The export form may contain mandatory and optional fields. The data entered in the export form will be included in the XML-metadata accompanying the media files, so that it can be processed by subsequent modules or systems.