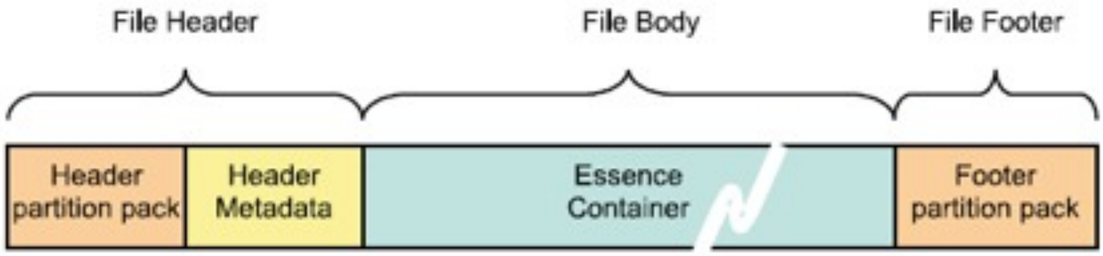
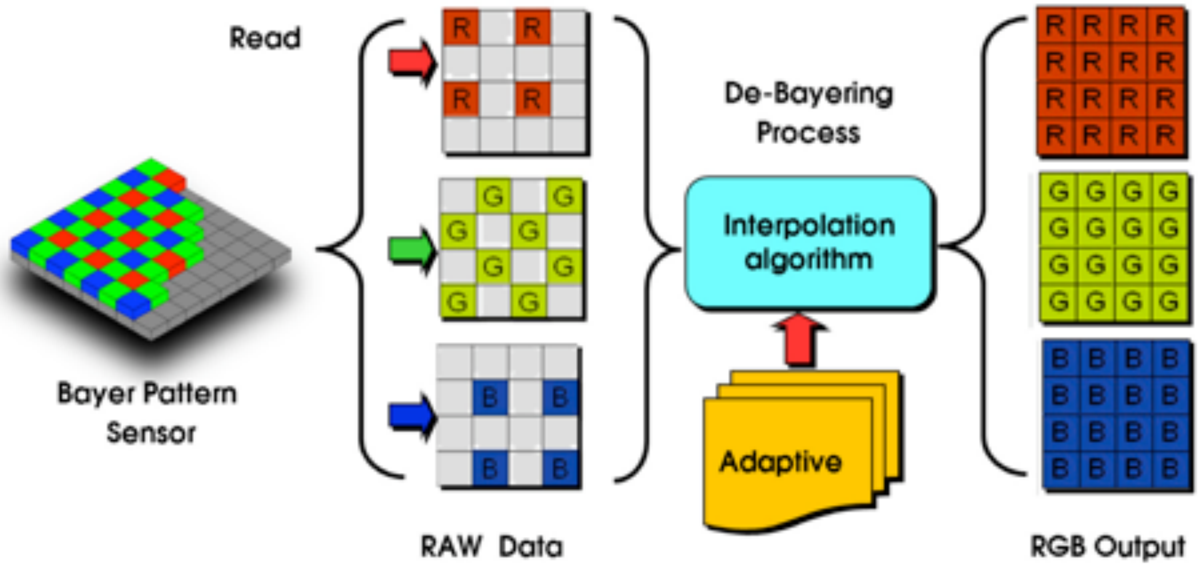
Wrappers, Essence and Codecs
Pre-Post and Post-Post or the Postman Knocks

Wrappers(containers):
AVI, MOV, MXF, DIVX, FLV, MP4, transport stream, program stream etc. As the term implies something (the encoded essence) is getting wrapped. How is this done and why is it important to know the details? When you take the wrapper off it looks similar to this: (http://www.pro-mpeg.org/documents/mxfwp2.pdf page 3)



If it is so simple why do we need so many different wrappers? Historically manufacturers have tried to lock in there customer base by using proprietary wrappers (.mov for Apple and .avi for Microsoft). There are also technical differences concerning the payload possibilities, the metadata facilities, and interleave implementations.
Interleave defines how the various payload components (audio, video, subtitles, etc.) are packed together. Given that we need to present audio, video, etc simultaneously (and want to transport them as a single file) it is easiest to mix them up so that the size of the buffer at the presentation device can be reduced. Otherwise we would have to buffer the whole audio portion before we could start playback. Interleaving 6 chunks of video with a total duration of 1 second and 1 second of audio divided into 4 chunks results in this, V0 A0 V1 A1 V2 V3 A2 V4 A3 V5, interleave pattern. This way there is no perceptible delay and the size of the receive buffer is kept to a minimum. Shorter blocks mean that the synchronization overhead is higher, longer blocks mean that the receive buffer has to be larger.
Payload possibilities are limited by the variety of different media which may be embedded, for example an .avi does not natively support the inclusion of subtitles, and the number of codecs supported. Most containers still do not support 3D using the MultiViewCodec. Menu support similar to DVD is available in an .m2ts but not in a .flv.
Metadata of some kind is included in all wrappers as this is required in order to instruct the player application which codec to use. Playback implementation is also much simpler when information regarding length and format are included. Unfortunately if this embedded metadata is inaccurate certain applications will simply refuse to process the file, on the other hand applications like Quicktime will play back an .mxf file even though it does not conform 100% to spec.There are applications, such as Avisynth, which will rewrap a file on the fly, but be careful as this is one way the metadata can get corrupted. If you have a file that has a corrupted header tools like Mencoder may be able to help. You may be able to patch the header because you know what the file is supposed to be, or an automatic fix on the basis of duplicate information in the header may be possible. These two methods do not require accessing every frame in the file and therefore can be quite quick, the third method requires reindexing the file and depending on the processing power available can be excruciatingly slow to the point where a recapture is faster.
Container formats: http://en.wikipedia.org/wiki/Comparison_of_container_formats

Essence (payload):
If you have an .mxf file utilizing the jpeg2000 video codec the essence may be 2K, 4K, etc at 4:4:4 or 4:2:2, interlaced or progressive, with a specific color space and pixel aspect ratio. We have not included frame rate as though a sequence may be recorded at a specific Fps (and this rate stored in the metadata) it may be played back at any rate the presentation device supports. Different pixel aspect ratios are used in digital video formats for PAL(1.07), NTSC(0,9)and anamorphic versions PAL(1.42), NTSC(1,2) as well as for different flavours of HD. Again this has to be stored in the metadata as it is difficult or impossible to automatically extract this information from the essence. Remember, when ingesting material via Ethernet from a VTR there is no switch for 16:9 vs 4:3, thus the technical metadata taken from the deck will not include the pixel aspect ratio.
4:4:4 or 4:2:2 do not refer to the number of samples per picture but only to the relationship between the samples of RGB or YUV etc. The actual sample rates used in the production environment range from 8 to 24bit and these may be expressed per color or per pixel. A 4:2:2 YUV may be sampled at 8 bits per color but then be referred to as 16 bit. This is done to make the math easier when determining how many bits are required to represent a particular raster ie 720x576x16 will tell me that I need 829440 bytes per frame (CCIR601 PAL).
YUV formats fall into two distinct groups, the packed formats where Y, U (Cb) and V (Cr) samples are packed together into macropixels which are stored in a single array, and the planar formats where each component is stored as a separate array, the final image being a fusing of the three separate planes. Similarly RGB frames may be stored from the top down or the bottom up. With the advent of RAW formats there is a now a third way in which the picture representation may be stored. The sensor in the camera has a particular pattern of RGB elements, “Bayer” being the most common, the data array reflects the order in which these elements are read out.



Today things can really get mixed up because production will quite frequently require mixing different essences in order to achieve a completed program. At one end of the scale is UHDTV at 7680 × 4320 (we are leaving out architectural still cameras and scientific imaging which have even higher resolution), at the other end is QCIF at 176x144.

Codec (compresson scheme):

Does an uncompressed file need a codec? Well, if you have an uncompressed file with V210 essence from Blackmagic Design you cannot open it in your NLE with an AJA capture card ,which also makes V210 uncompressed files, unless you first install the proper codec. So a codec not only defines how data reduction is implemented it also carries instructions for playback of the file.

As the above example illustrates some "transcoding" operations are simpler than others. If you have a software application that is taking an excessive amount of time to do a simple rewrap then either your settings are wrong ie the resulting essence is different, or you should use a different application.

The purpose of compression is to achieve a data rate reduction with no quality loss. Production codecs are either lossless or lossy. If you take your original file then subtract your encoded file and the result is pure black (no residual information) then you have a lossless codec. Lossy codecs are usually measured by the number of decode/encode cycles which can take place before a certain level of degradation occurs. The SMPTE in 1998 (SMPTE/EBU Task Force for Harmonized Standards for the Exchange of Program Material as Bitstreams Final Report: Analyses and Results**)** determined that 7 cycles with a quality loss of under 12.5% was required for a production format. Whether this is referred to as "visually lossless" or "virtually lossless" depends on the codec manufacturer. Codecs such as Prores, DNXHD, HQX, Cineform, SR-lite, etc. purport to meet this requirement. These are commonly referred to as intermediate codecs and are intraframe ie each frame is encoded separately. During production we are not really concerned with which codec will be used for delivery as each presentation platform, be it IPTV to a large screen or streaming over a WLAN for an Ipad, has different requirements. What does concern us is whether the codec we are using can easily be converted into any delivery format required without an additional transcode.

Pre-Post we need to be certain that our files are intact and ready to be used. Post-Post we need to be sure that our wrapper conforms to spec and the codec and essence are what is required for simple high quality conversion. Otherwise the Postman will knock and someone will have to pay the bill.