

## MPEG-2/MPEG-4 System 기술

서울시립대학교

전자전기컴퓨터공학부

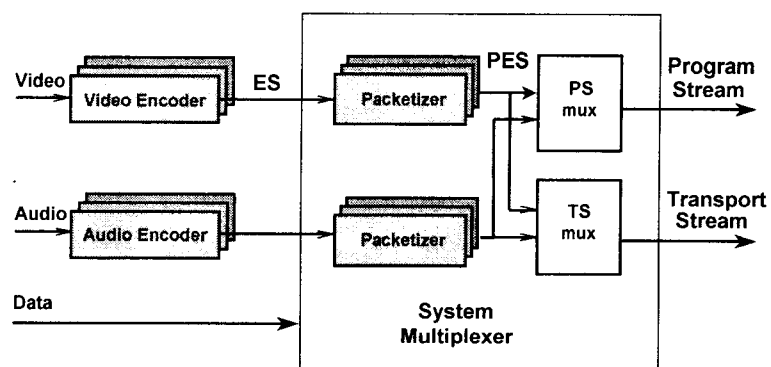
김용한 ([yhkim@uoscc.uos.ac.kr](mailto:yhkim@uoscc.uos.ac.kr))

## MPEG-2 Systems

## Parts of MPEG-2 Standard

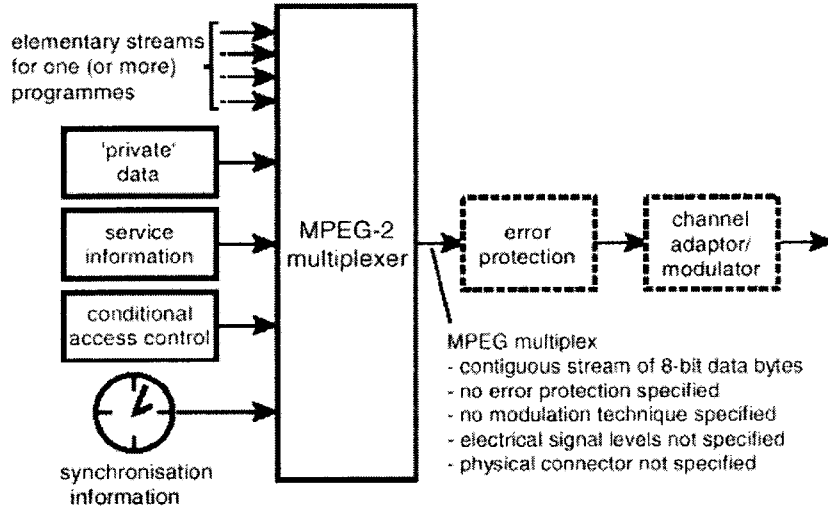
- ✓ Part 1: Systems
- ✓ Part 2: Video
- ✓ Part 3: Audio
- ✓ Part 4: Conformance testing
- ✓ Part 5: Reference software
- ✓ Part 6: DSM-CC extensions
- ✓ Part 7: Advanced audio coding (AAC)
- ✓ Part 9: Real-time Interface
- ✓ Part 10: Conformance testing for DSM-CC extensions

## Main Functionality



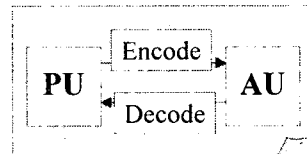
ES : Elementary Stream  
 PES : Packetized Elementary Stream  
 PS : Program Stream  
 TS : Transport Stream

## Details on the encoder side

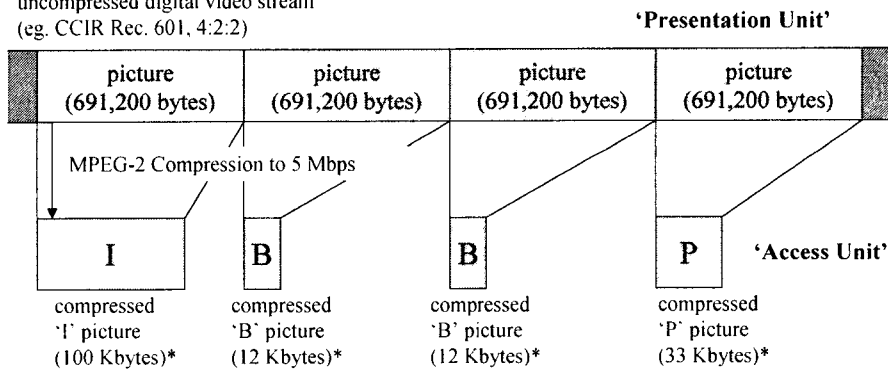


## Elementary Stream (1)

Access Unit vs. Presentation Unit



uncompressed digital video stream  
(eg. CCIR Rec. 601, 4:2:2)



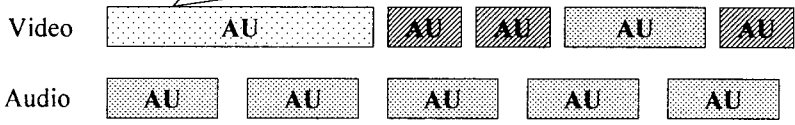
\* The actual size depends on target bit-rate and complexity of picture

## Elementary Stream (2)

AUs can have timing information.

PTS or both PTS and DTS

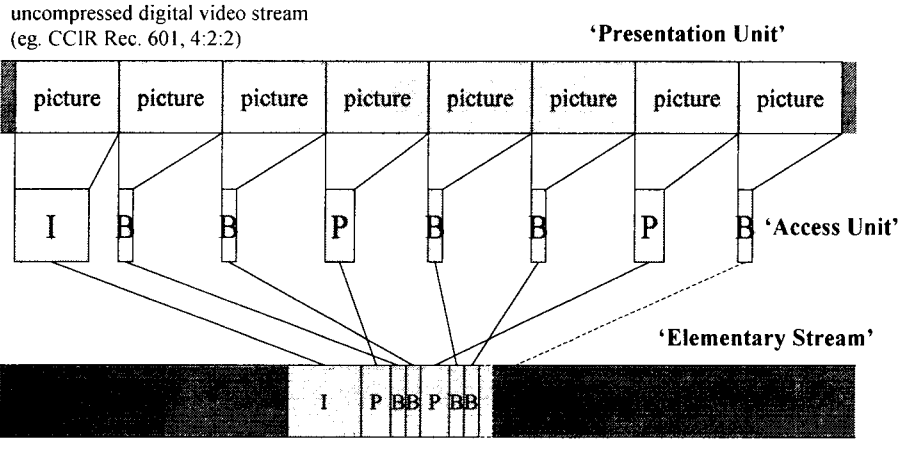
An access unit is the smallest data entity to which timing information can be attributed  
(타이밍 정보가 부가될 수 있는 가장 작은 단위)  
(MPEG2 : frame or field, MPEG4 : VOP)



Data from each elementary stream are multiplexed together with information that allows synchronized presentation of the elementary streams within the program  
(각 ES의 데이터는 그 ES의 synchronized presentation 정보를 가지고 multiplexed 된다)

## Elementary Stream (3)

Creating an ES from AUs

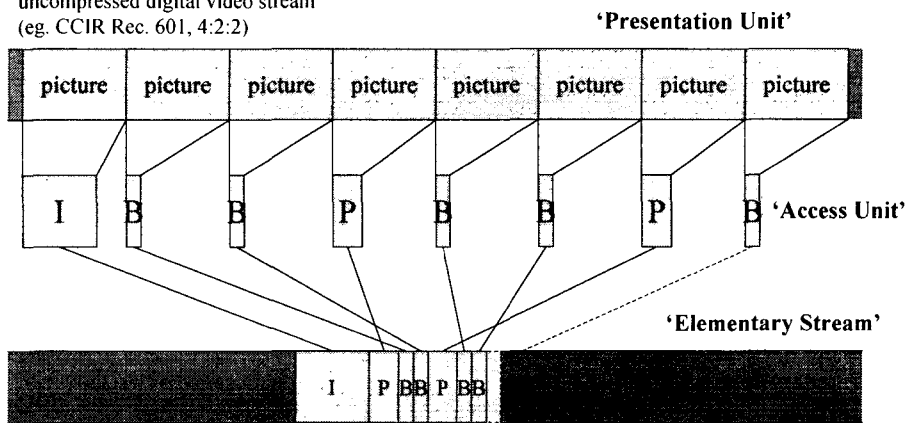


(Optionally, an ES can have either ESCR or ES\_Rate.)

## Elementary Stream (ES) (3)

(Creating an ES from AUs)

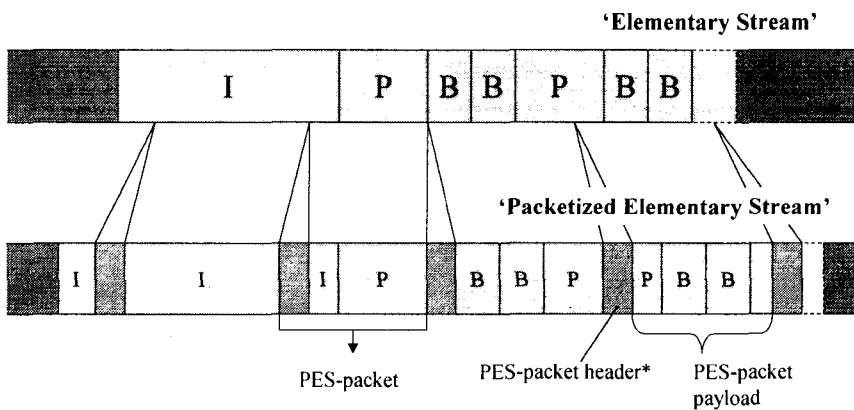
uncompressed digital video stream  
(eg. CCIR Rec. 601, 4:2:2)



(Optionally, an ES can have either ESCR or ES\_Rate.)

## Packetized Elementary Stream (1)

(Creating a PES from an ES)



\* The PES packet header is right where timing information (an PTS or both PTS and DTS) and optionally, either ESCR or ES\_Rate can locate themselves.

### Packetized Elementary Stream (3) (The types of ES' which PES packets can have)

stream id	Common	stream_id	Stream coding
8	0x000001		packet_start_code_prefix
		bc	program_stream_map
		bd	private_stream_1
		be	padding_stream
		bf	private_stream_2
		c0-df	audio_stream
		e0-ef	video_stream
		f0	ECM_stream
		f1	EMM_stream
		f2	ISO/IEC 13818-6 DSMCC_stream
		f3	ISO/IEC 13522_stream
		f4	ITU T Rec. H.222.1 type A
		f5	ITU T Rec. H.222.1 type B
		f6	ITU T Rec. H.222.1 type C
		f7	ITU T Rec. H.222.1 type D
		f8	ITU T Rec. H.222.1 type E
	f9	ancillary_stream	
	fa-fe	Reserved data stream	
	ff	program_stream_directory	

### Packetized Elementary Stream (4) (PES packet header syntax)

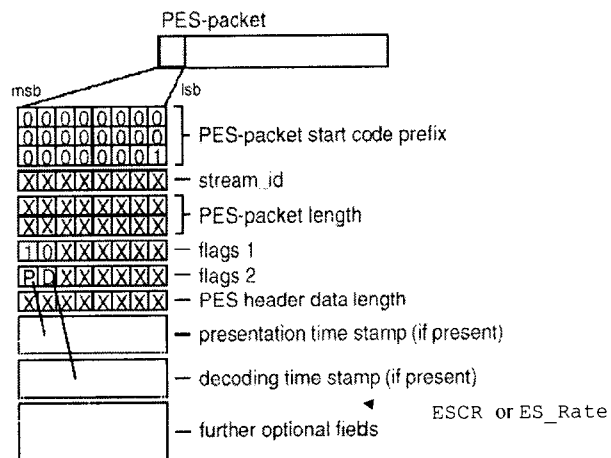
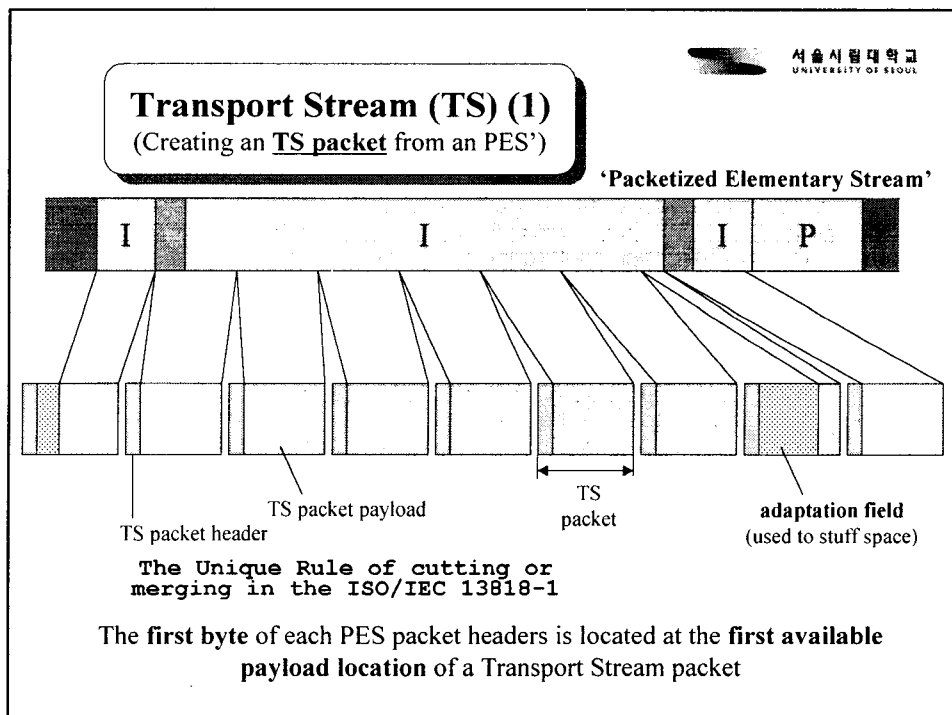
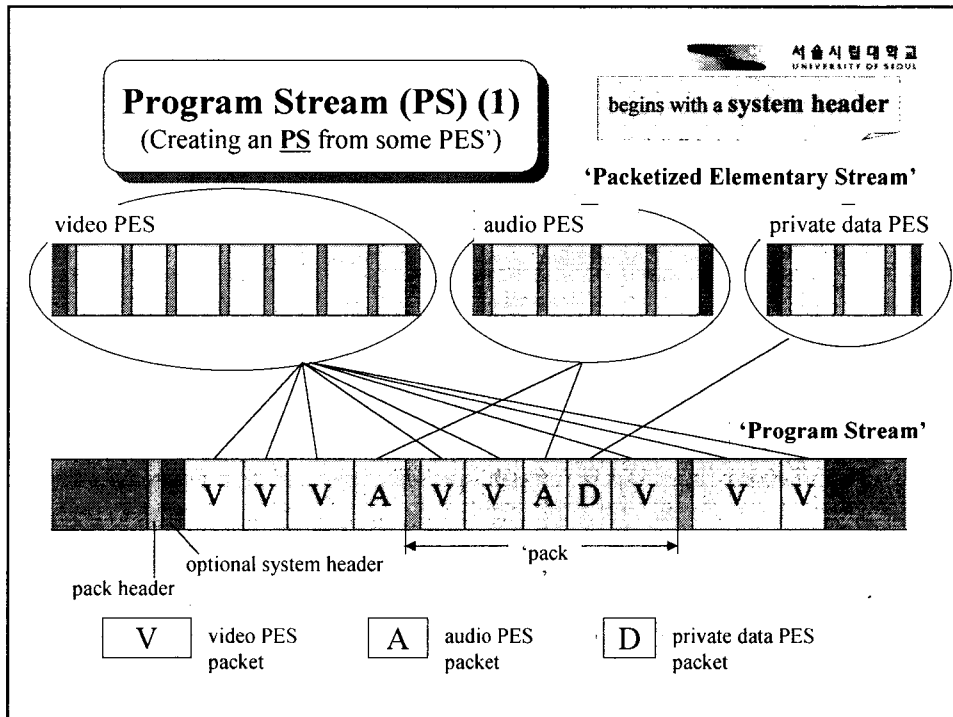
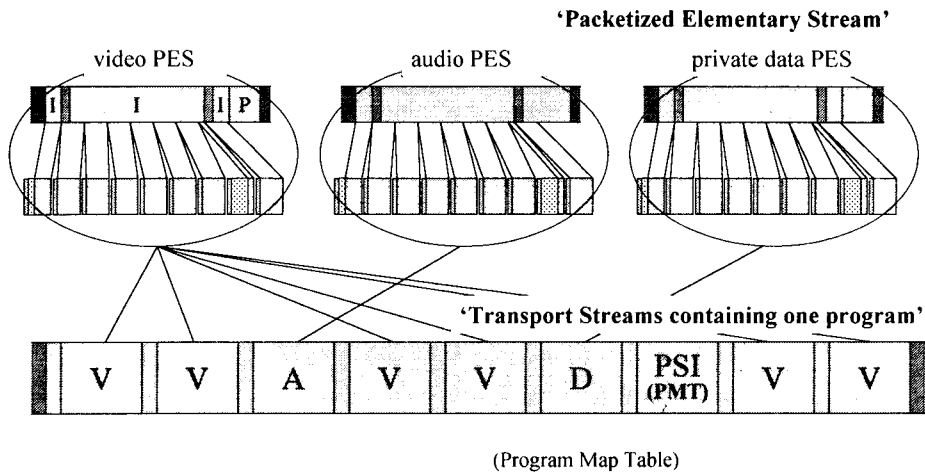


Fig. 4 - A PES-packet header.



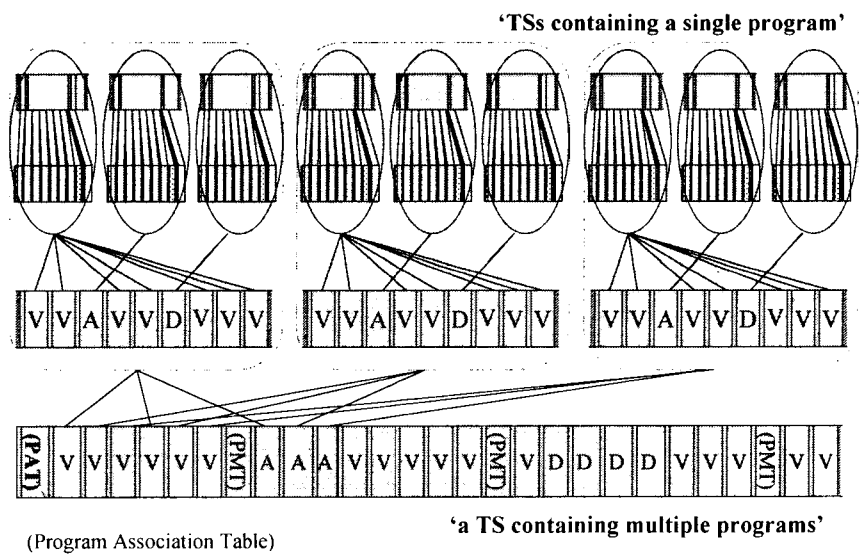
## Transport Stream (TS) (2)

(Creating an **TS containing one program** from some TS packets)



## Transport Stream (TS) (3)

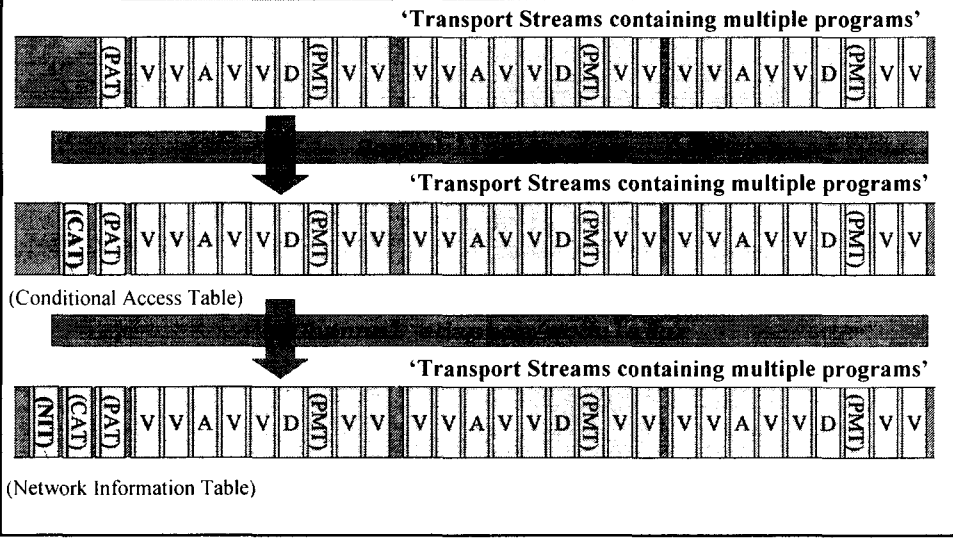
Creating a **TS containing multiple programs**





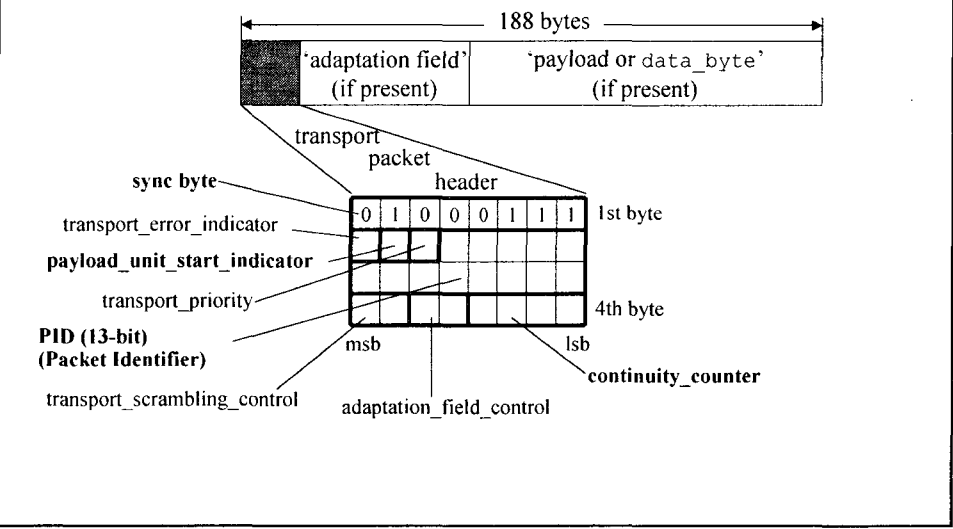
## Transport Stream (TS) (4)

(Adding some PSI tables to TS)



## Transport Stream (TS) (5)

(The structure of TS packet header)



## Transport Stream (TS) (6) (PID)

### Packet ID (PID)

- identifies, via the PMT(`stream_type`), the contents of the data contained in the Transport Stream packets
- ▶ ex) Transport Stream packets of one PID value carry data of one and only one elementary stream (only in case PES packet data)
- ▶ 1 PID  $\subset$  1 ES (only in case PES packet data)



`stream_id` (PS::PES)

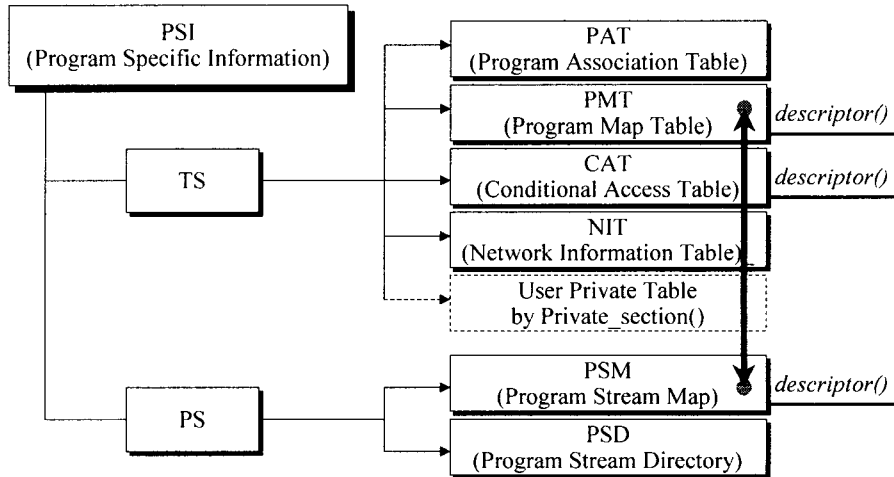
## Transport Stream (TS) (7)

(The types of data which the payload of a TS packet can have)

(괄호 안의 숫자는 `stream_type`)

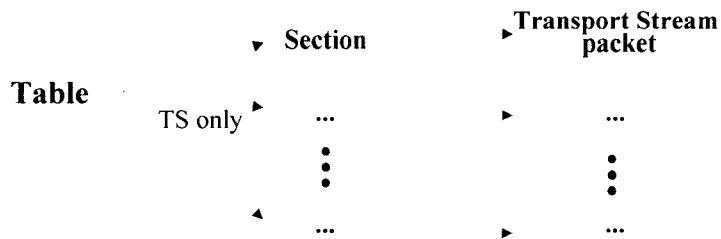
- data\_byte (payload)**
- ▶ PES packet data (including private data (0x06))
  - ▶ PSI sections (including `private_section` (0x05))
  - ▶ PSI sections + packet stuffing bytes
  - ▶ User private (0x80~0xff)
  - ▶ any value (null packet, PID = 0x1fff)
  - ▶ Nothing !!! (`adaptation_field_control` = '10')

## Program Specific Information (PSI) (1) (all kinds of PSI)



## Program Specific Information (PSI) (2) (A table is split up into sections to transport packet data)

Scrambling을 하면 안됨. 단 private\_section의 private\_data\_byte는 제외

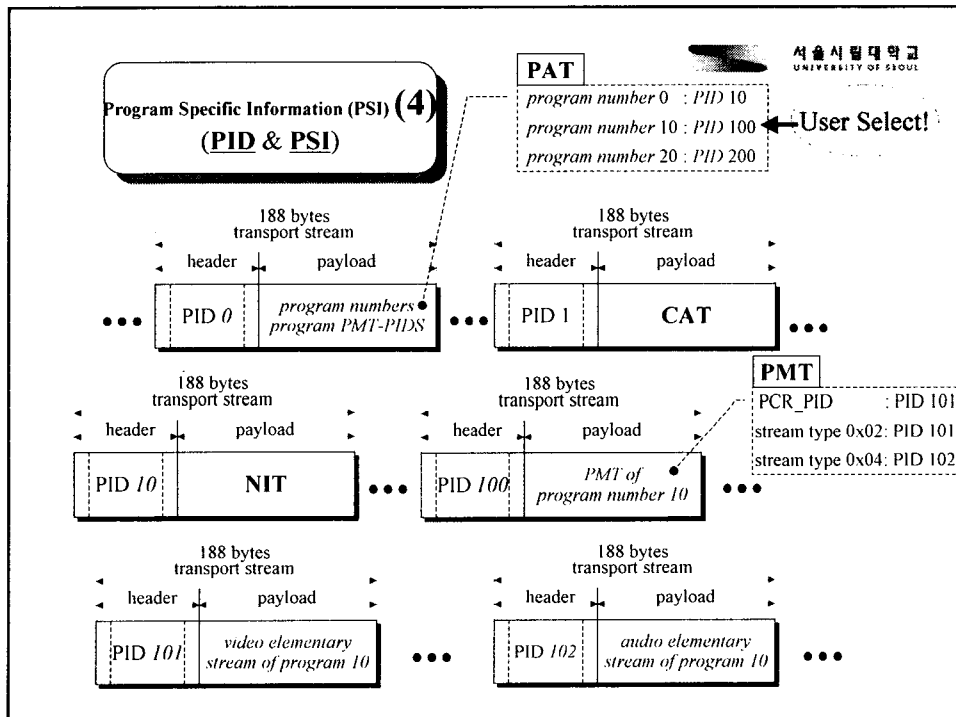


- PSI sections  $\leq 1024$  bytes  $\gg$  188 bytes (TS packet)
- private\_sections  $\leq 4096$  bytes
- payload\_unit\_start\_indicator  $\xrightarrow[\text{PSI}]{1}$  pointer\_field

## Program Specific Information (PSI) (3) (PID Table)

Value	Description
0x0000	Program Association Table
0x0001	Conditional Access Table
0x0002-0x000f	Reserved
0x0010-0x1ffe	May be assigned as network_PID, program_map_PID, elementary_PID, or for other purposes
0x1fff	Null packet

NOTE - The transport packets with PID values 0x0000, 0x0001, and 0x0010-0x1ffe are allowed to carry a PCR



## Program Specific Information (PSI) (5) (stream type assignment)

stream type  
8

Value	Description
0x00	ITU-T   ISO/IEC reserved
0x01	ISO/IEC 11172-2 Video
0x02	ITU-T Rec. H.262   ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x03	ISO/IEC 11172-3 Audio
0x04	ISO/IEC 13818-3 Audio
0x05	ITU-T Rec. H.222.0   ISO/IEC 13818-1 private sections
0x06	ITU-T Rec. H.222.0   ISO/IEC 13818-1 PES packets containing private data
0x07	ISO/IEC 13522 MHEG
0x08	Annex B - DSM CC
0x09	ITU-T Rec. H.222.1
0x0a	ISO/IEC 13818-6 type A
0x0b	ISO/IEC 13818-6 type B
0x0c	ISO/IEC 13818-6 type C
0x0d	ISO/IEC 13818-6 type D
0x0e	ISO/IEC 13818-1 auxiliary
0x0f	ISO/IEC 13818-7 Audio with ADTS transport syntax
0x10-0x7f	ITU-T Rec. H.222.0   ISO/IEC 13818-1 reserved
0x80-0xff	User Private

## Time Stamps (1) (the two kinds of time stamps)

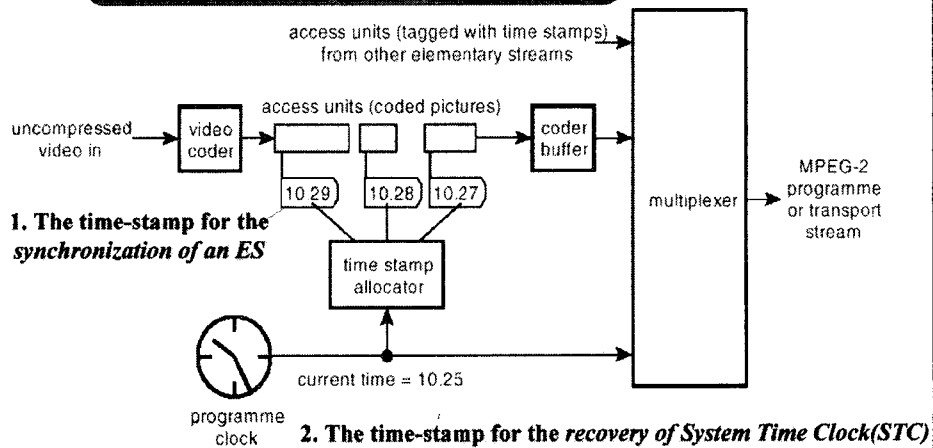
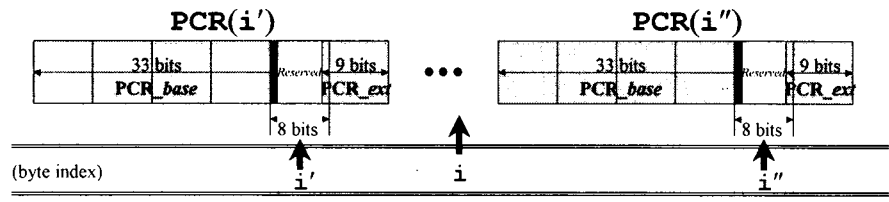


Fig. 15 - Time stamps are allocated to access units by the multiplexer.  
(Note the allowances made to access unit times for the decoder clock).

### Time Stamps (2)

(to calculate a bit-rate between PCR or SCR)

for only one program



- $PCR(i') = \text{system\_color\_frequency} \times t(i')$  ◀ 실세계 시간  
 $= PCR\_base(i') \times 300 + PCR\_ext(i')$   
 $PCR\_base(i') = PCR(i') / 300$   
 $PCR\_ext(i') = PCR(i') \% 300$
- $\text{transport\_rate} = \frac{i'' - i'}{t(i'') - t(i')}$  ◀ PCR구간 내에서는 일정하다고 봄      바이트수 / 시간
- $t(i) = t(i') + (i - i') / \text{transport\_rate}$

### Decoder-Side

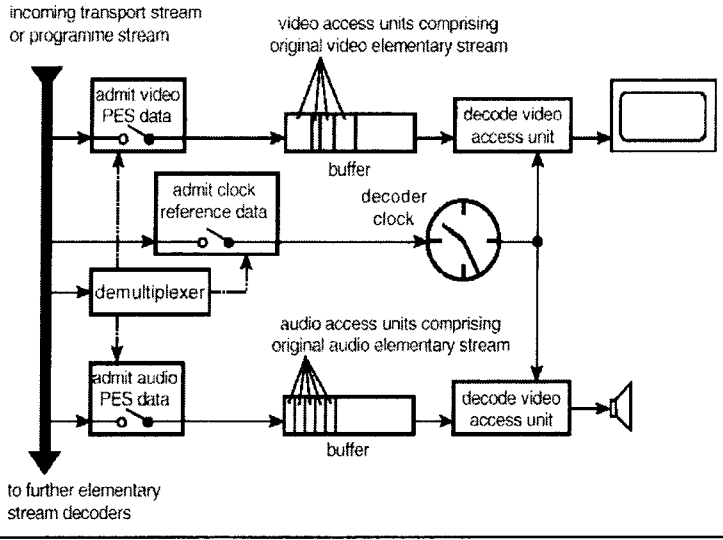


Fig. 14 - A possible decoder.

# MPEG-4 System

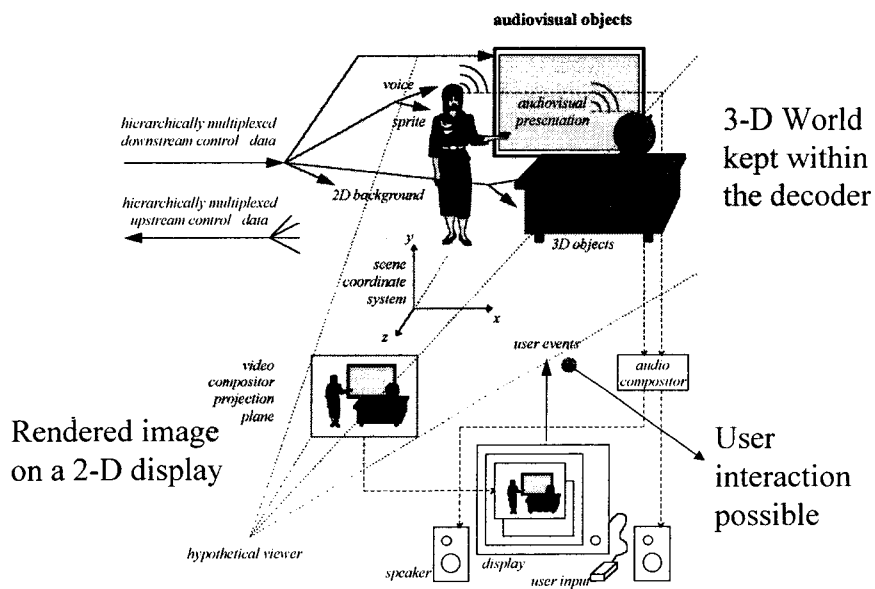
## Parts of MPEG-4 Standard

- 14496-1 Systems
- 14496-2 Visual
- 14496-3 Audio
- 14496-4 Conformance Testing
- 14496-5 Reference Software
- 14496-6 Delivery Multimedia Integration Framework (DMIF)

## What's New in MPEG-4

- ✓ **Object-based coding**
- ✓ **Synthetic/Natural Hybrid Coding(SNHC)**
- ✓ Flexible Decoder Architecture
  - ☞ Tool-based
  - ☞ S/W-based implementation
- ✓ Error Resilience
- ✓ Flexible Composition/Rendering
- ✓ NOT RESTRICTED TO “Very-low bit rate coding”
- ✓ DMIF

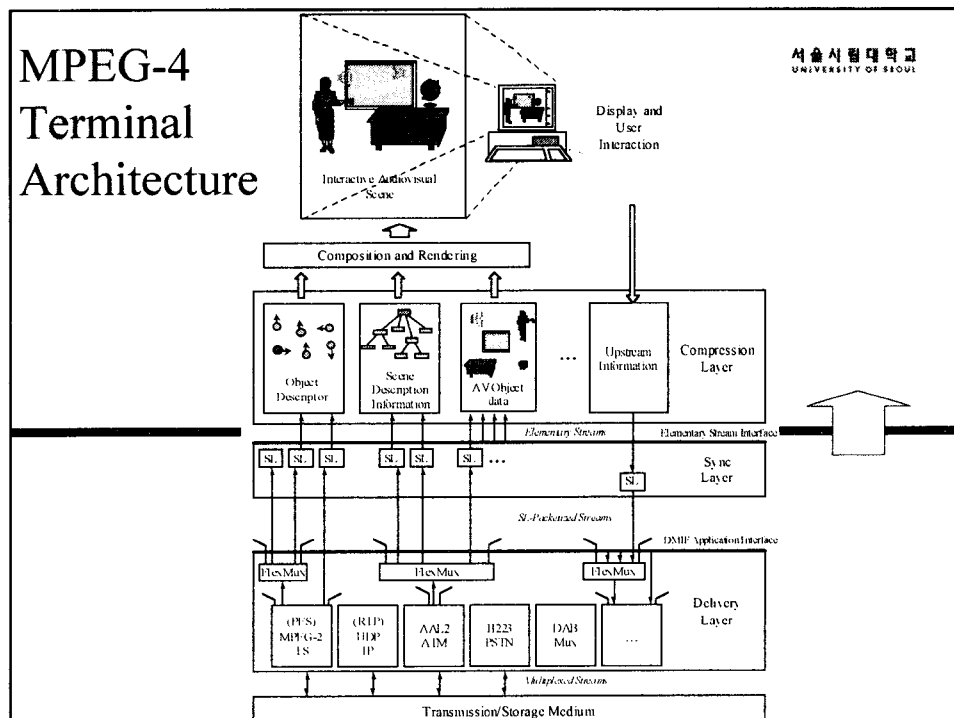
## Scene composition and presentation of AV objects

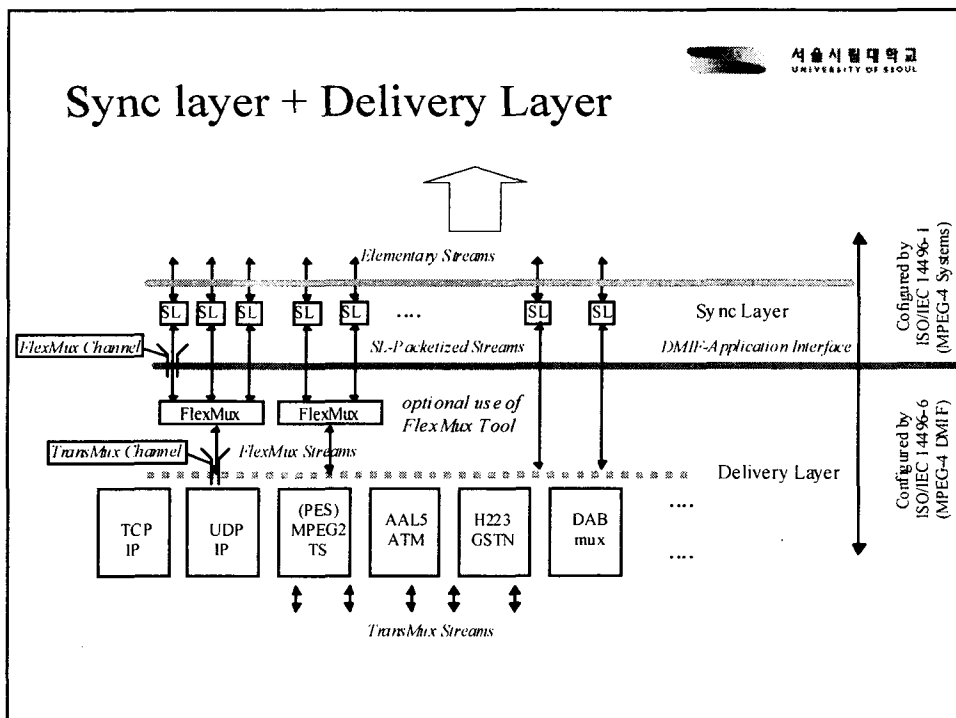
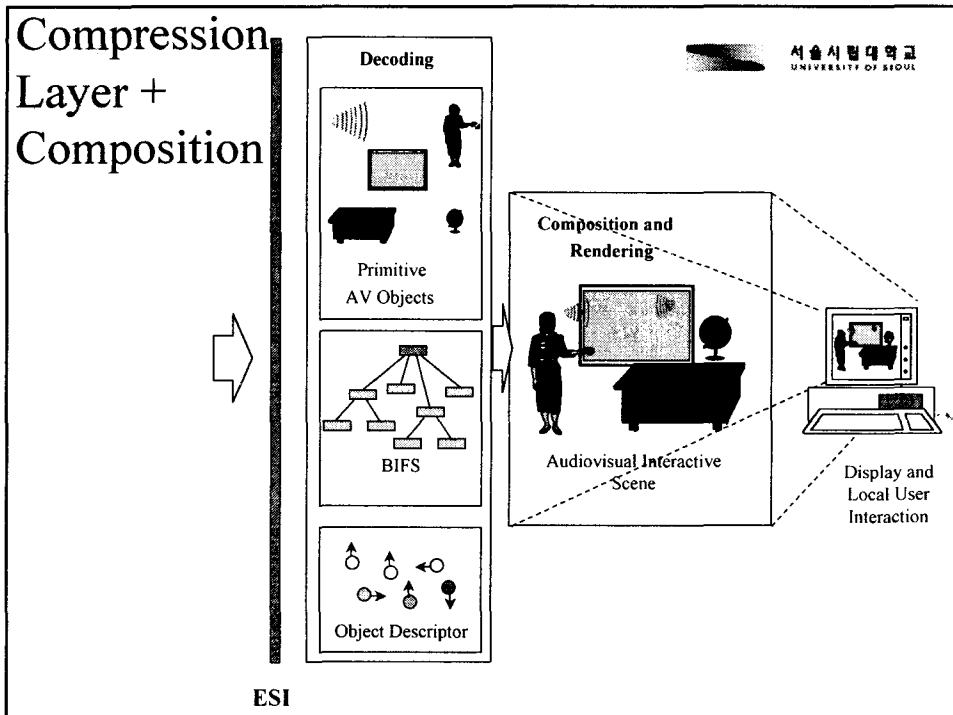




## MPEG-2 vs MPEG-4 Systems

- ✓ Video v.s. Video Object (VO)
- ✓ Picture v.s. Video Object Plane (VOP)
- ✓ Examples
  - ☞ Program Clock Reference (PCR) v.s. Object Clock Reference (OCR)
  - ☞ Elementary Stream : Video v.s. VO
  - ☞ AU : Picture v.s. VOP

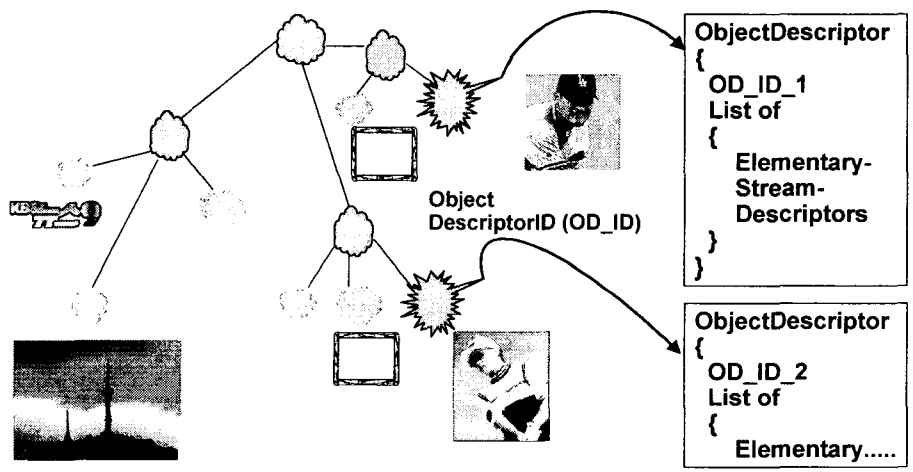
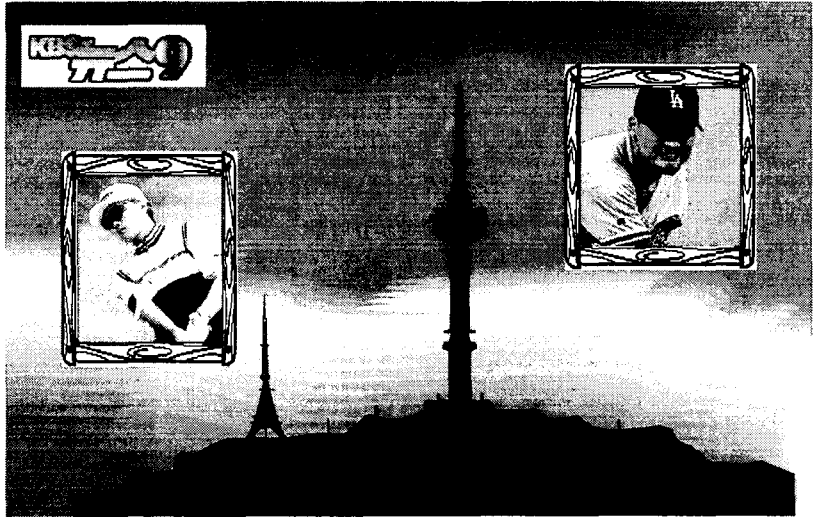


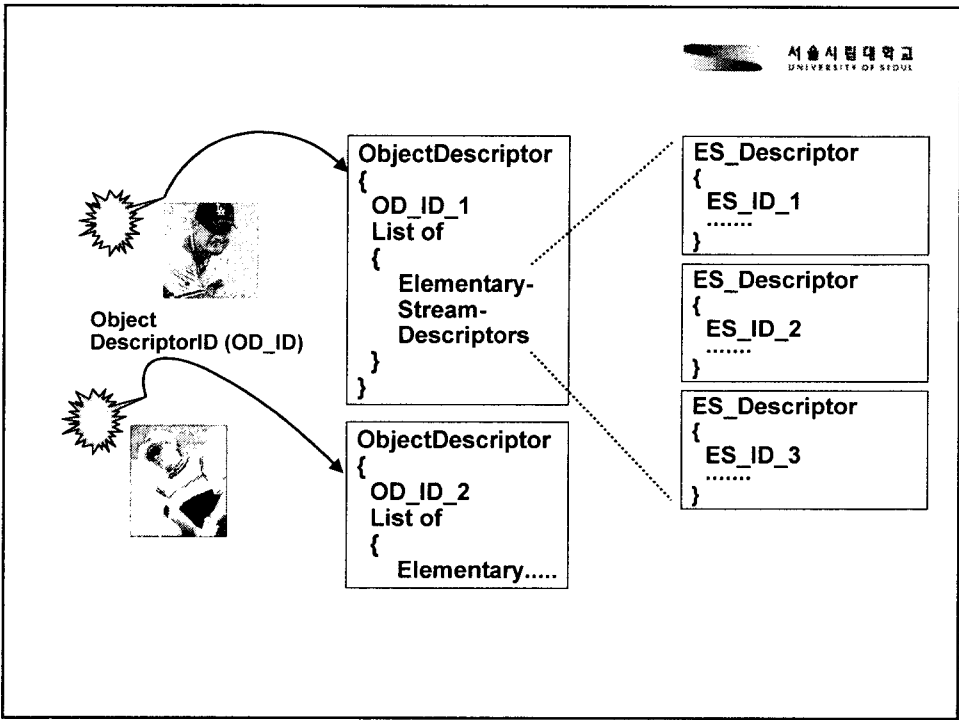


## IS 14496-1 Contents (Version 1)

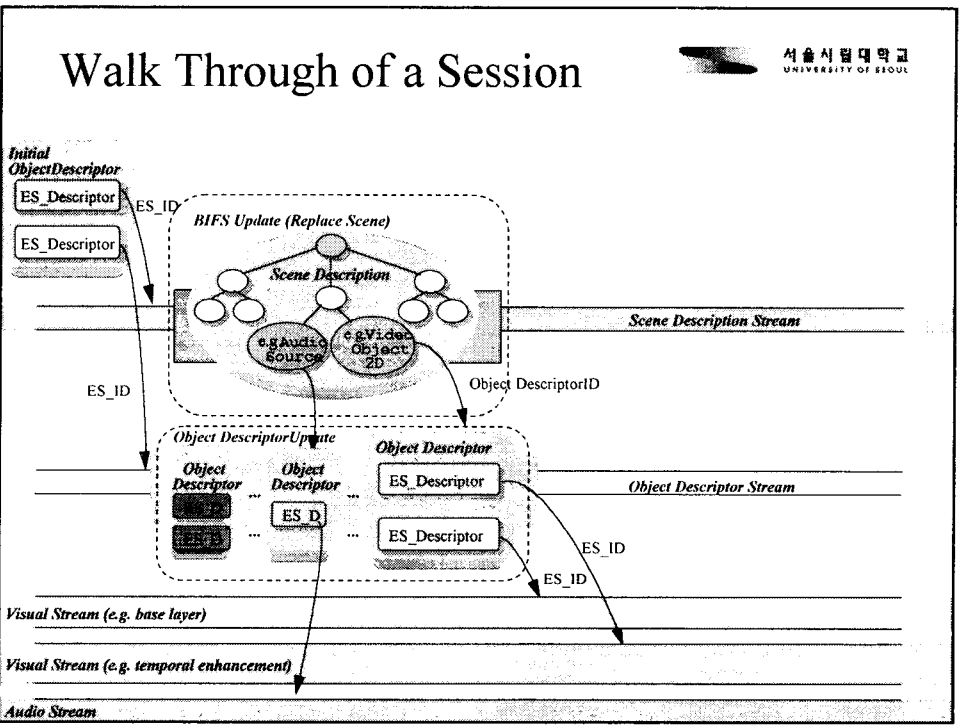
- ✓ System Decoder Model
- ✓ Object Description Framework
- ✓ Scene Description
- ✓ Synchronization Layer
- ✓ Multiplex Layer
- ✓ Syntactic Description Language (omit)
- ✓ Profiles

# Object Description Framework





## Walk Through of a Session

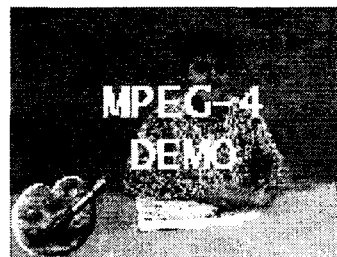


# Scene Description

## 2D examples

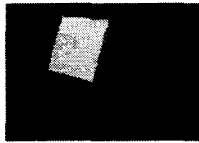


Animated Text+ Video + Still Images

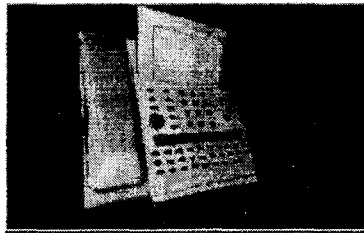


Video Overlay of Animated Text+ Video + Still Images

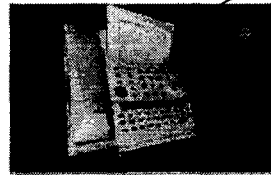
## 3D examples



3D Graphic Primitives

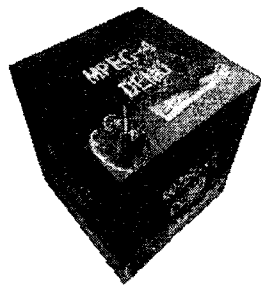


Complex 3D Mesh

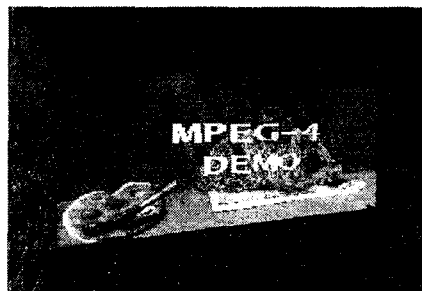


Scene with Face Object

## 2D/3D Examples

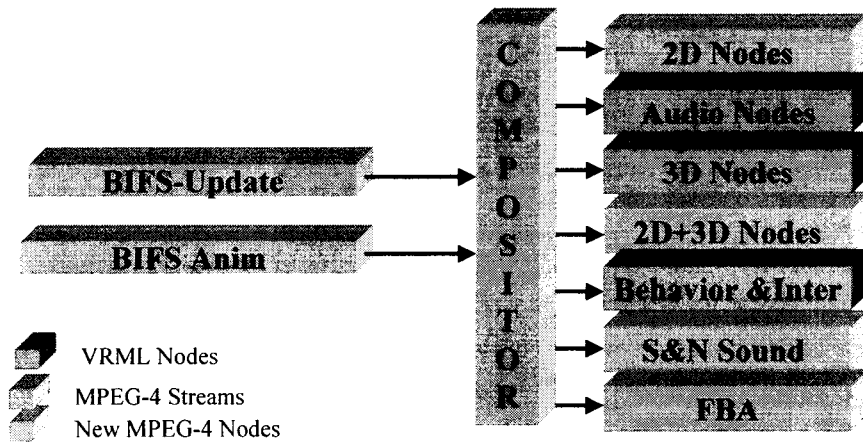


2D or 3D scene as a texture map on 3D



2D inside a 3D plane

## BIFS Components



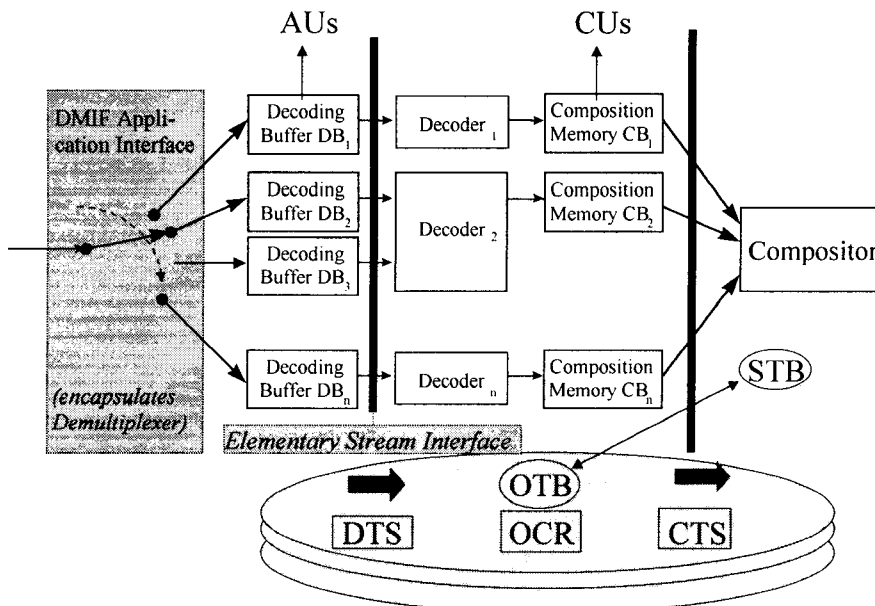
## System Decoder Model



## Purpose of System Decoder Model (SDM)

- ✓ an abstract view of the behavior of a terminal complying with ISO/IEC 14496.
- ✓ used by the sender to predict how the receiving terminal will behave in terms of buffer management and synchronization when decoding data received in the form of elementary streams
- ✓ “Push Model”

## Systems Decoder Model



## AU and CU

- ✓ Elementary Stream (ES)
  - ☞ A consecutive flow of mono-media data from a single source entity to a single destination entity on the compression layer
  
- ✓ Access Unit (AU)
  - ☞ An individually accessible portion of data within an *elementary stream*. An access unit is the smallest data entity to which timing information can be attributed
  
- ✓ Composition Unit (CU)
  - ☞ An individually accessible portion of the output that a decoder produces from access units

## Time Base, Time Stamp, Clock Reference

- ✓ Time Base
  - ☞ The notion of a clock; it is equivalent to a counter that is periodically incremented
  
- ✓ Time Stamp
  - ☞ An indication of a particular time instant relative to a time base
  - ☞ A sampled value of the counter
  
- ✓ Clock Reference
  - ☞ A special kind of periodic time stamps that is used by a decoder to recover the time base of the encoder

## STB and OTB

- ✓ System Time Base (STB)
  - ☞ The time base of the terminal.
  - ☞ Its resolution is implementation-dependent.
  
- ✓ Object Time Base (OTB)
  - ☞ A time base valid for a given elementary stream, and hence for its decoder
  - ☞ The resolution of this OTB can be selected as required by the application or as defined by a profile

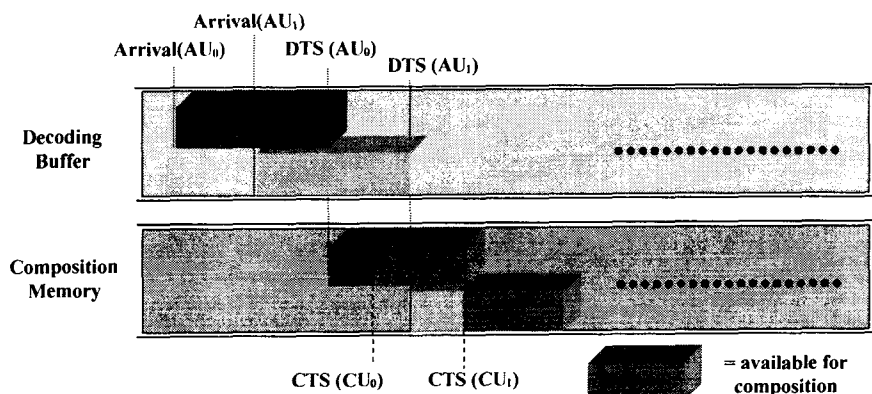
## OCR

- ✓ Object Clock Reference (OCR)
  - ☞ A clock reference that is used by a decoder to recover the time base of the encoder of an elementary stream

## DTS and CTS

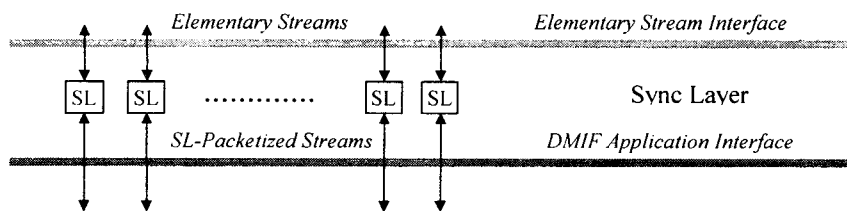
- ✓ Composition Time Stamp (CTS)
  - ☞ An indication of the nominal composition time of a composition unit
  
- ✓ Decoding Time Stamp (DTS)
  - ☞ An indication of the nominal decoding time of an access unit
  - ☞ shall only be conveyed for an access unit that carries a composition time stamp as well, and only if the DTS and CTS values are different

## Synchronization (2 AU Example)



# Sync layer

## SL-Packet



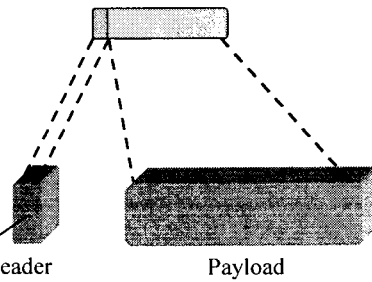
✓ Sync Layer packet (SL-Packet)

- ☞ The smallest data entity managed by the sync layer consisting of a configurable header and a payload. The payload may consist of one complete access unit or a partial access unit

## Functionality of Sync. Layer

- ✓ Boundaries of AccessUnit. Access Units may use more than one SL-Packet.
- ✓ Provides consistency checking for lost packets.
- ✓ Carries Object Clock Reference.
- ✓ Carries Decoding and Composition time stamps.

Sync. Layer Packet



Maximally configurable : very little overhead

## Notes on Sync. Layer

- ✓ An SL packet does not contain an indication of its length.
  - Therefore, SL packets must be framed by a suitable lower layer protocol, e.g., the FlexMux tool.
  - Consequently, an SL-packetized stream is not a self-contained data stream that can be stored or decoded without such framing.
- ✓ An SL-packetized stream does not provide identification of the ES\_ID associated to the elementary stream in the SL packet header.
  - This association must be conveyed through a stream map table using the appropriate signalling means of the delivery mechanism.

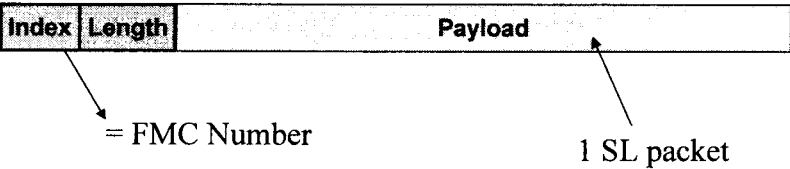
# Multiplexing

## FlexMux (Optional)

- ✓ Low complexity multiplexing tool (interleaving of data)
  - ☞ low overhead (RTP/UDP/IP ⇒ high overhead)
  - ☞ low delay
  
- ✓ FlexMux packet : data from one or more SL packets
- ✓ FlexMux stream : a sequence of FlexMux packets interleaved into one stream
- ✓ FlexMux Channel (FMC) : Identification of ES
  - ☞ Each SL-packetized stream is mapped into one FlexMux Channel
  
- ✓ Framing and error detection should be provided by the transport protocol stack below FlexMux.

# FlexMux Modes

Simple Mode (1 FlexMux PDU)



MuxCode Mode (1 FlexMux PDU)

