# 이동 멀티미디어 응용을 위한 객체 기반 컨텐츠 서비스 기법

대구대학교
차 경 애
2005. 9. 2

# 목 차

- MPEG 기술의 소개

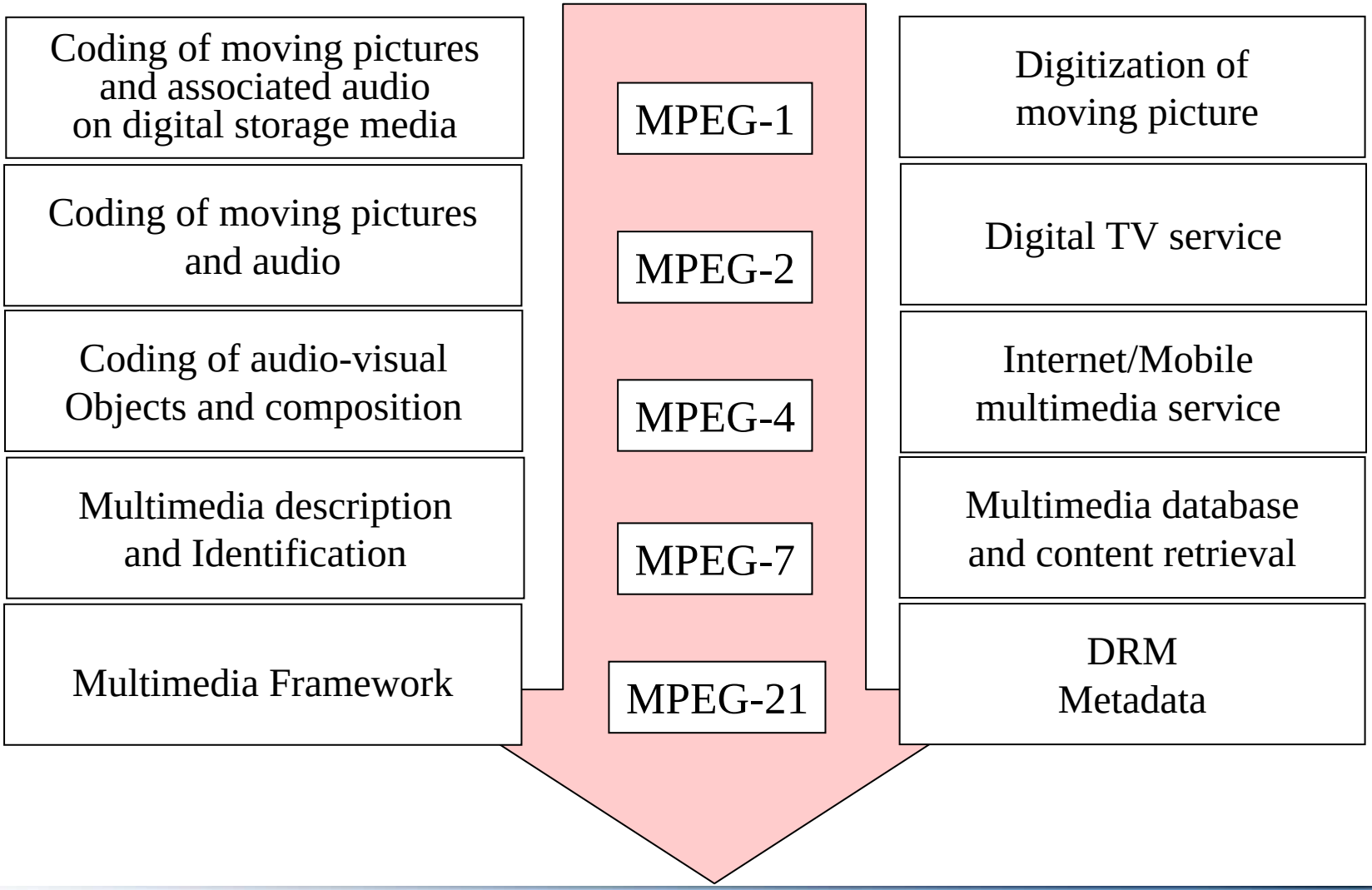
- MPEG-4 System
  - 객체 기반 멀티미디어 처리 기술
- 이동멀티미디어방송
  - 지상파 디지털 멀티미디어 방송 기술

- 요약

# MPEG

- Informal Title
  - Moving Picture Experts Group (MPEG)
- Formal Title
  - Coding of Moving Pictures and Audio
- Area of work
  - Development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio, and their combination, in order to satisfy a wide variety of applications.

# MPEG Technology Trends



# Accomplishments – MPEG-4(1/2)

- ISO/IEC 14496
- 1998, Coding of audio-visual objects
  - Part 1 – Systems (2001)
  - Part 2 – Visual (2001)
  - Part 3 – Audio (2001)
  - Part 4 – Conformance(2002)
  - Part 5 – Reference Software (2002)
  - Part 6 – DMIF – Delivery Multimedia Integration Framework (2000)
  - Part 7 – Optimized Software for MPEG-4 tools (2002)
  - Part 8 – MPEG 4 on IP framework (2002)
  - Part 9 – Reference Hardware (2003)
  - Part 10 – Advanced Video Coding (AVC) joint with ITU-T(2003)

# Accomplishments – MPEG-4(2/2)

- Part 11 – Scene Description and Application Engine
- Part 12 – ISO Base Media File Format
- Part 13 – IPMP Extensions
- Part 14 – MP4 File Format
- Part 15 – AVC File Format
- Part 16 – Animation Framework eXtension (AFX)
- Part 17 – Streaming text format
- Part 18 – Font compression and streaming
- Part 19 – Synthesized Streams
- Part 20 – Lightweight Application Scene Representation (LASeR)
- Part 21 – MPEG-F extension for rendering

# MPEG-4 – 특징

- Object based coding (Content-based, Arbitrarily shaped)
- MPEG-4 Systems에 의한 interactivity 제공
  - on each AV object
  - at the levels of coding, decoding, or objects composition
- Natural/Synthesized AV objects
  - natural video, 2D/3D graphics, text, …

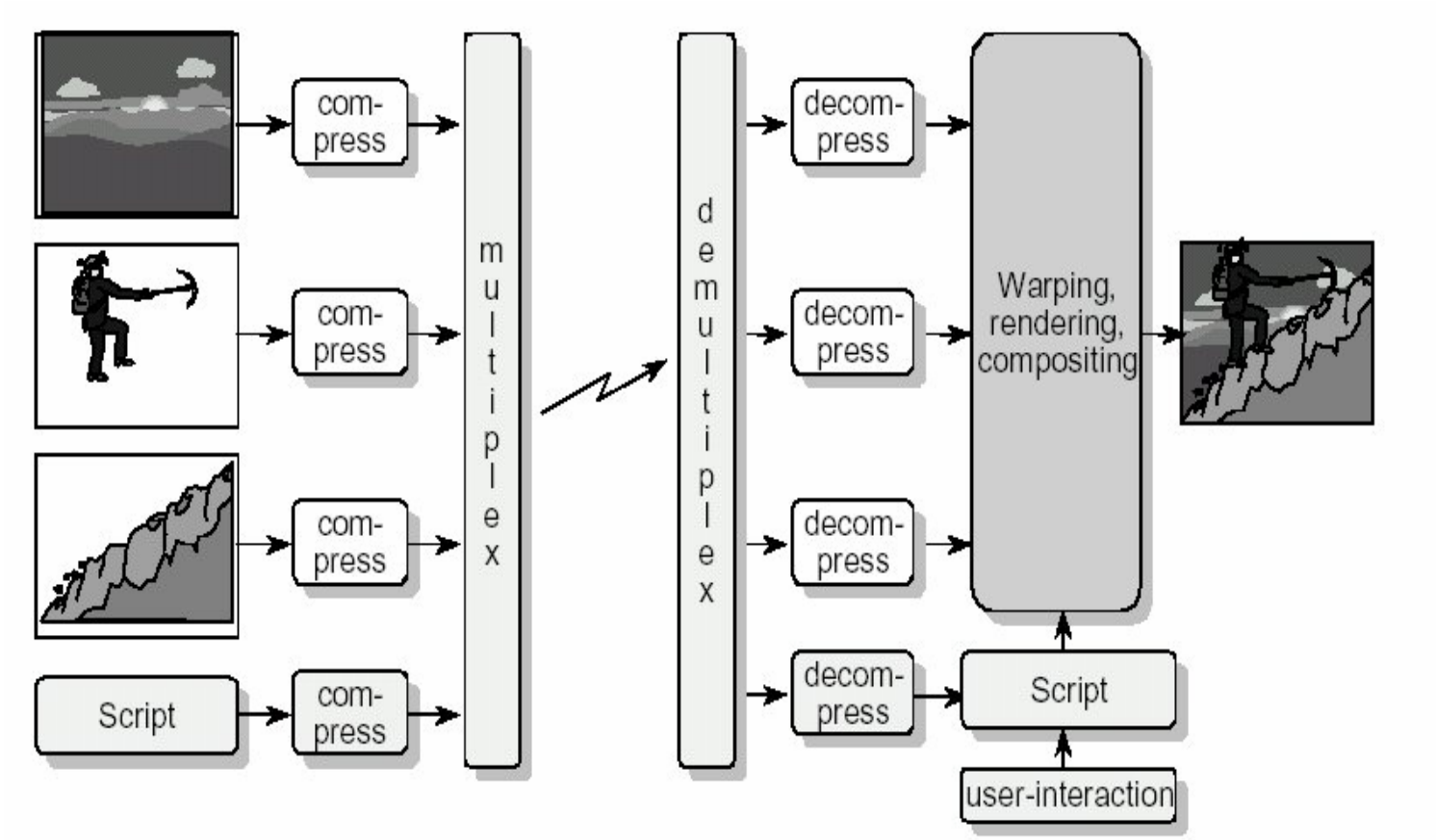
# MPEG-4 System

# MPEG-4 Principles

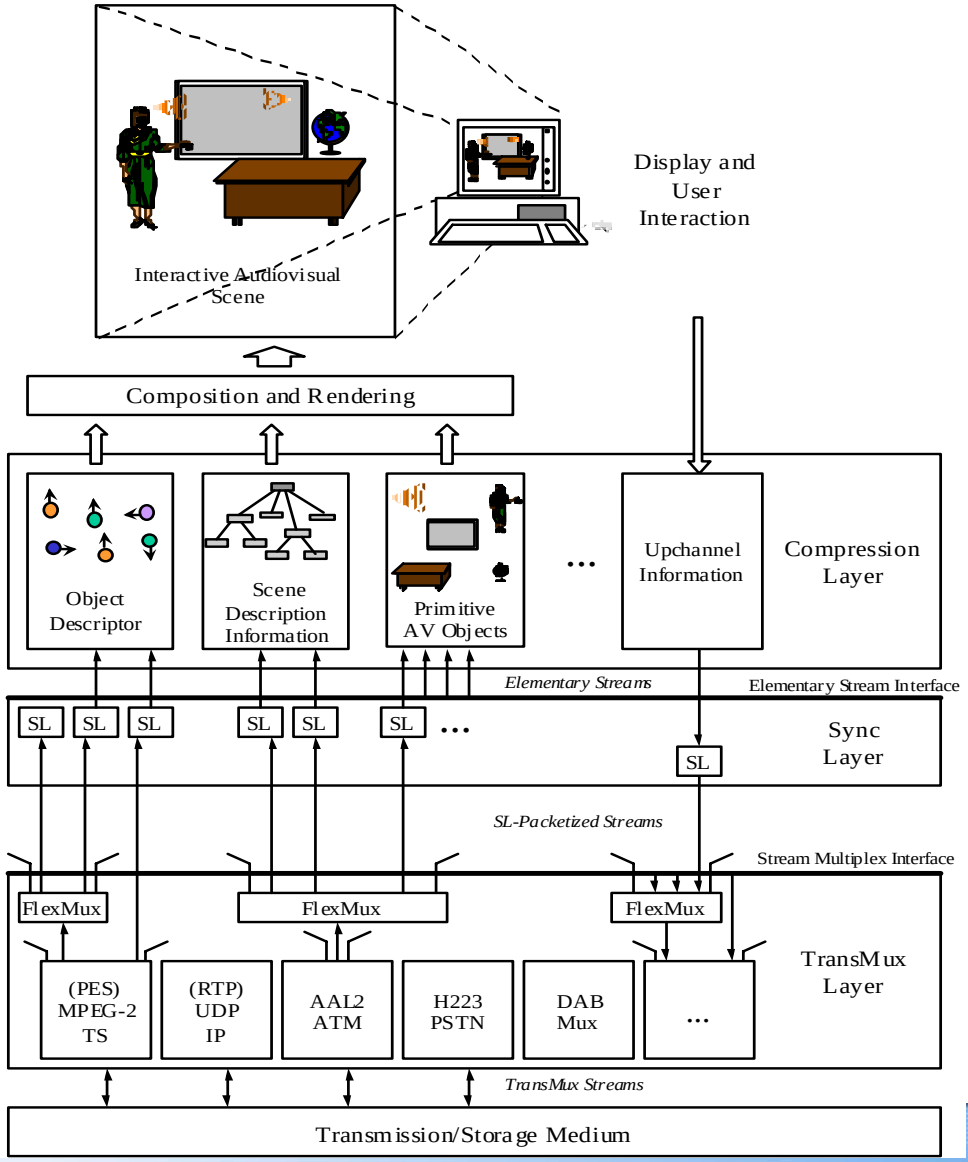
- Audio-visual scenes made of audio-visual objects composed together according to a scene description :
  - allows interaction with elements within the audio-visual scene,
  - coding scheme can differ for individual objects,
  - allows easy re-use of audio-visual content.

- Audio-visual objects can be of different nature :
  - audio (single or multi-channel) or video (arbitrary shape or rectangular),
  - natural (natural audio or video) or synthetic (text & graphics, animated faces, synthetic music),
  - 2D (Web like pages) or 3D (spatialized sound, 3D virtual world),
  - streamed (video movie) or downloaded (audio jingle)

# MPEG-4 Scene Concept



# MPEG-4 System Architecture




Processing Stage in Terminal

Compression Layer

Sync. Layer

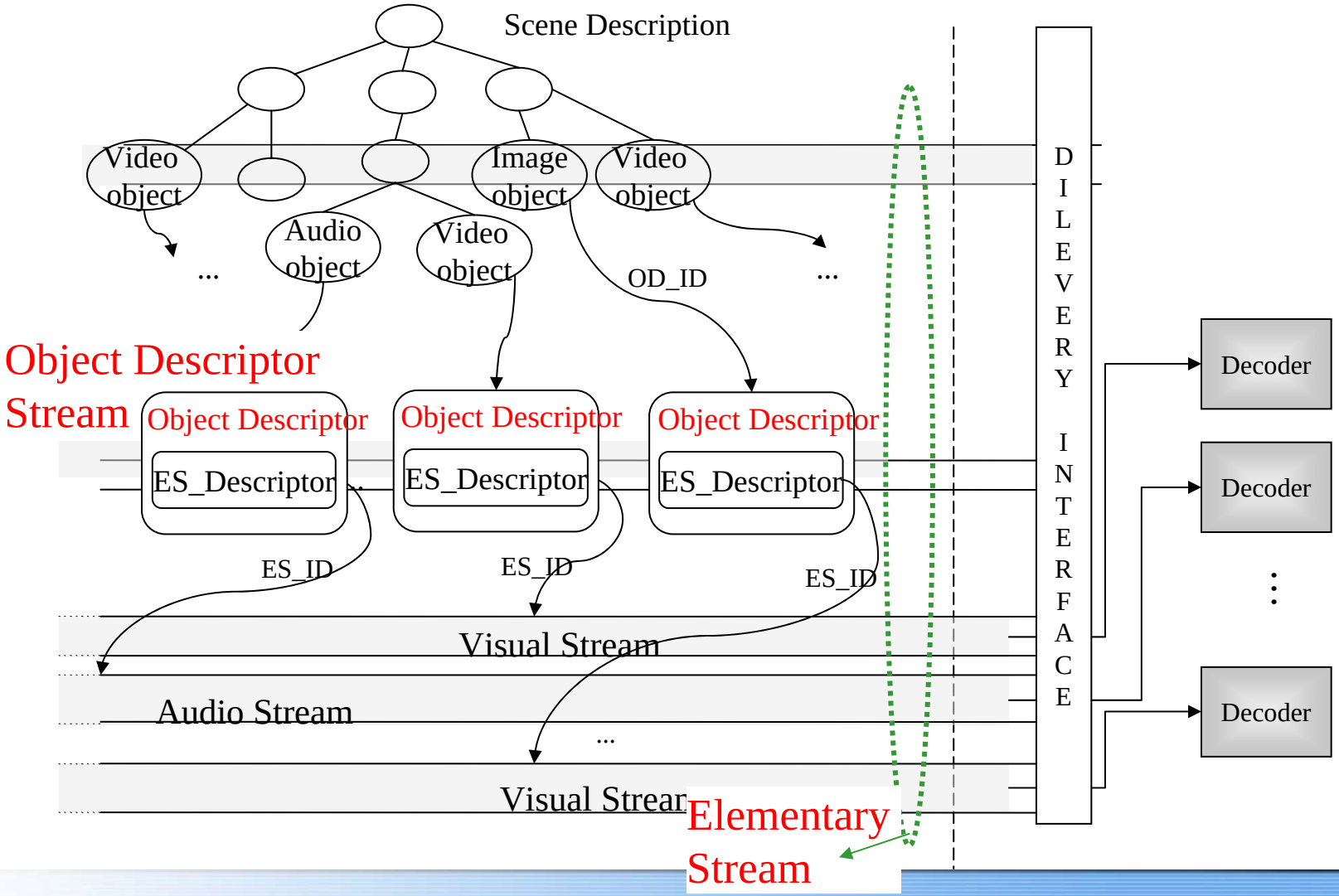
FlexMux/TransMux Layer

# Scene description

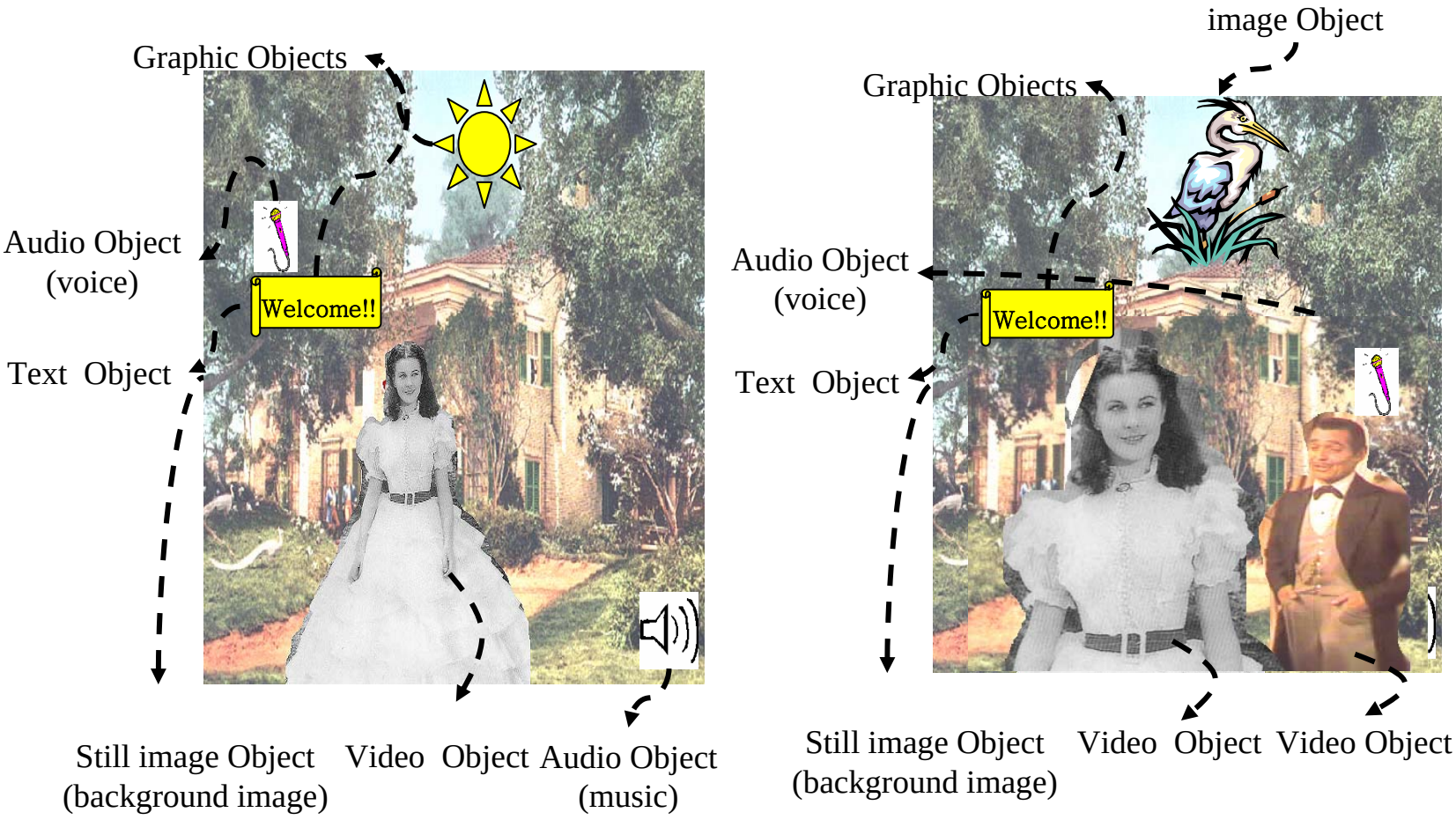
- BIFS :
  - Binary format for scene description
- Spatio-temporal specification of objects
- Objects behavior specification

```
Group {
   children [
      DEF Transform2D1000 Transform2D {
          translation -34.00 7.00
           scale 1.00 1.00
           rotationAngle 0.00
           children [
                    Shape {
                           appearance Appearance {
                                   texture ImageTexture {
                                                url 1
                                                …
                                  ]
                DEF Transform2D3001 Transform2D {
                                  translation 42.00 72.00
                                  scale 1.24 1.00
                                  rotationAngle 0.00
                                  children [
                                 Shape {
                                         appearance Appearance
{
                                                material DEF
}
                                                …
```

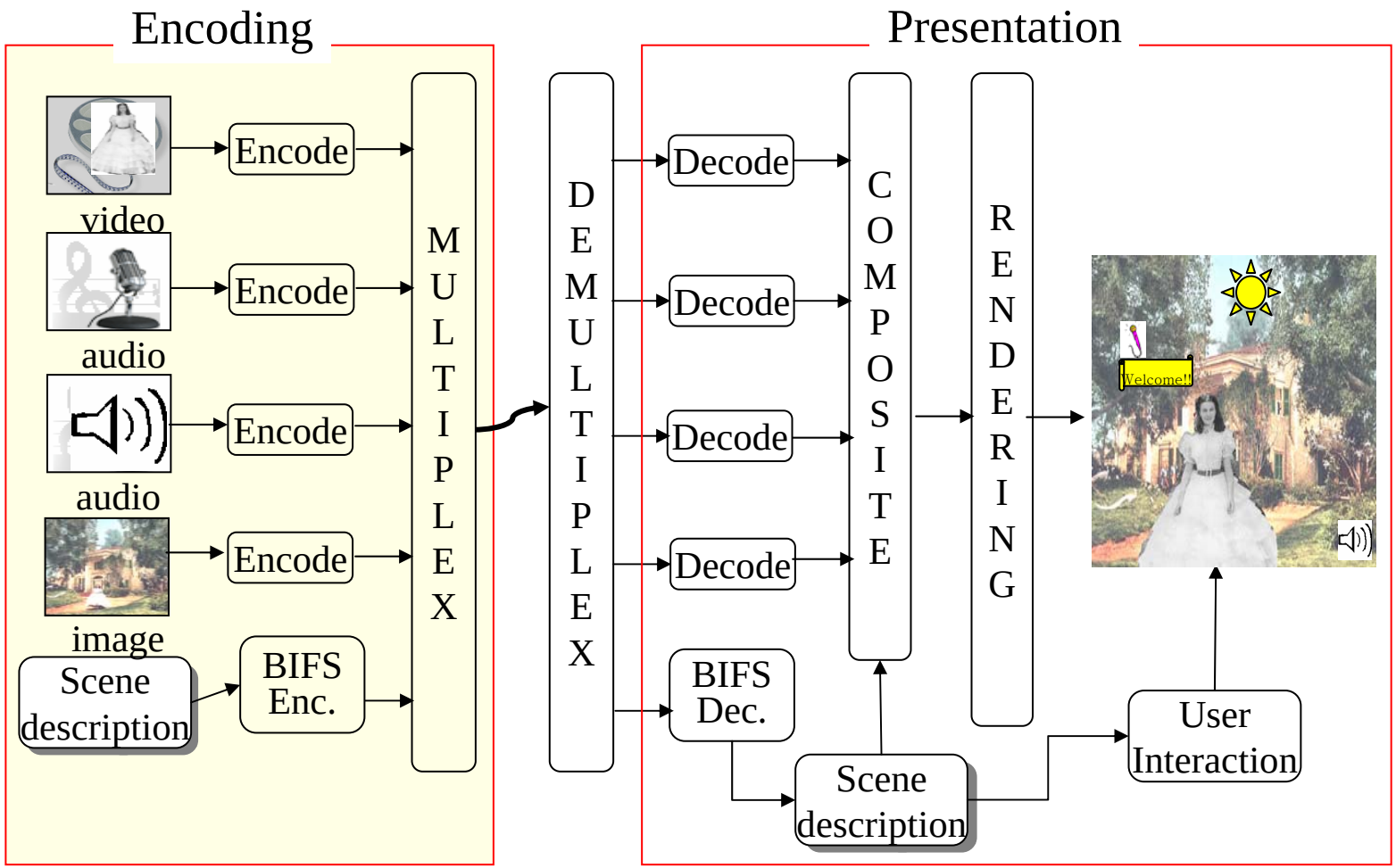
# Object descriptor & Elementary stream



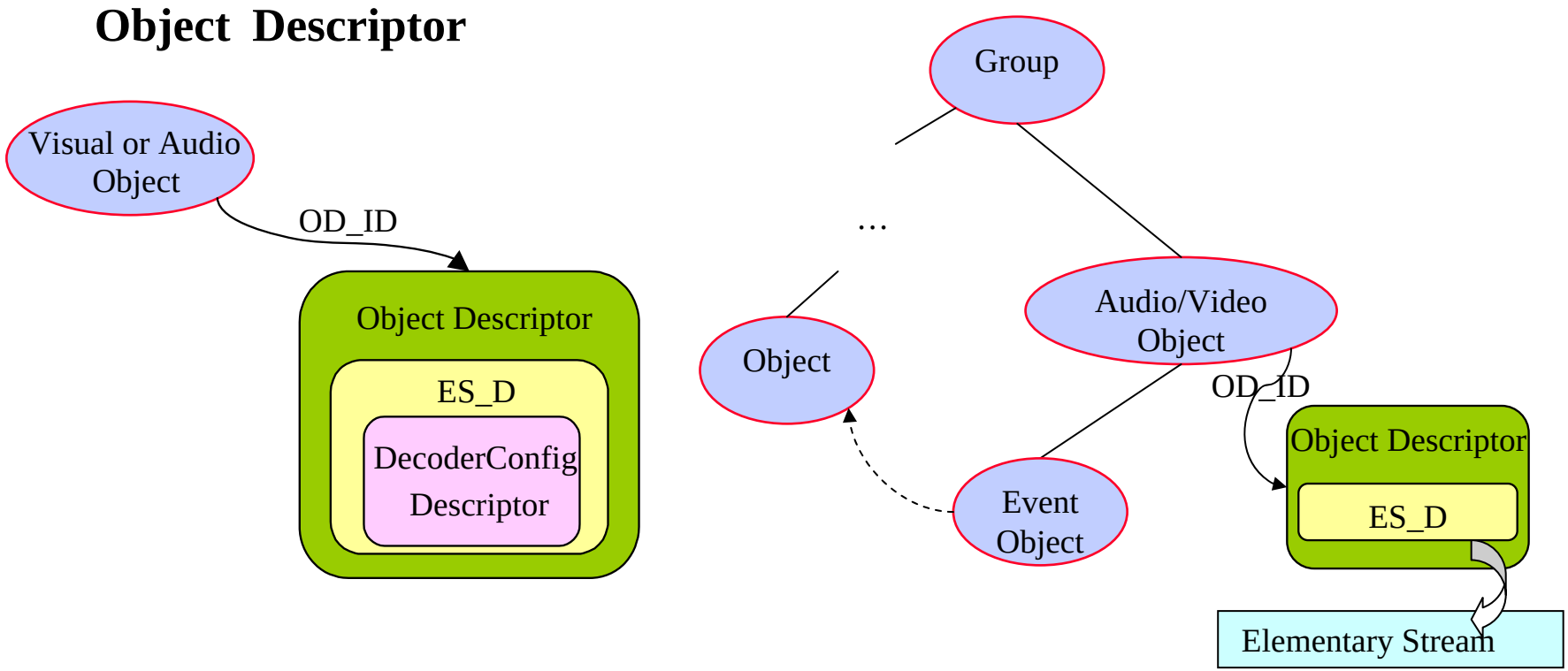
# An Example of MPEG-4 scene



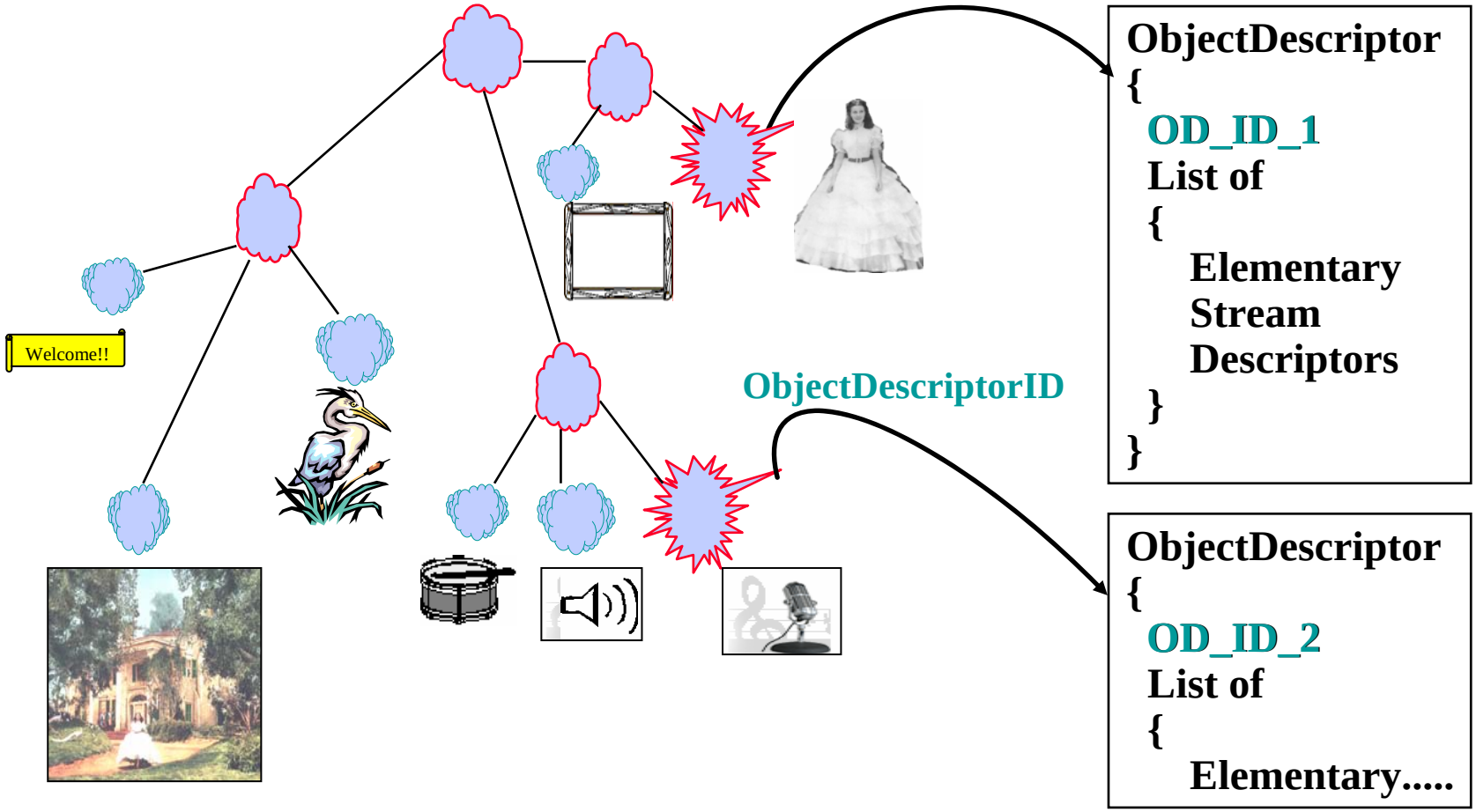
# Object-based compression and delivery



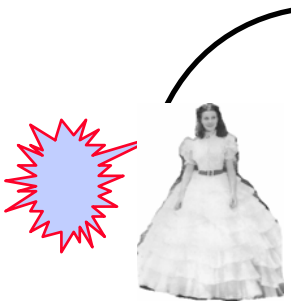
# Linking streams into the scene(1/6)



# Linking streams into the scene(2/6)



# Linking streams into the scene (3/6)



ObjectDescriptorID

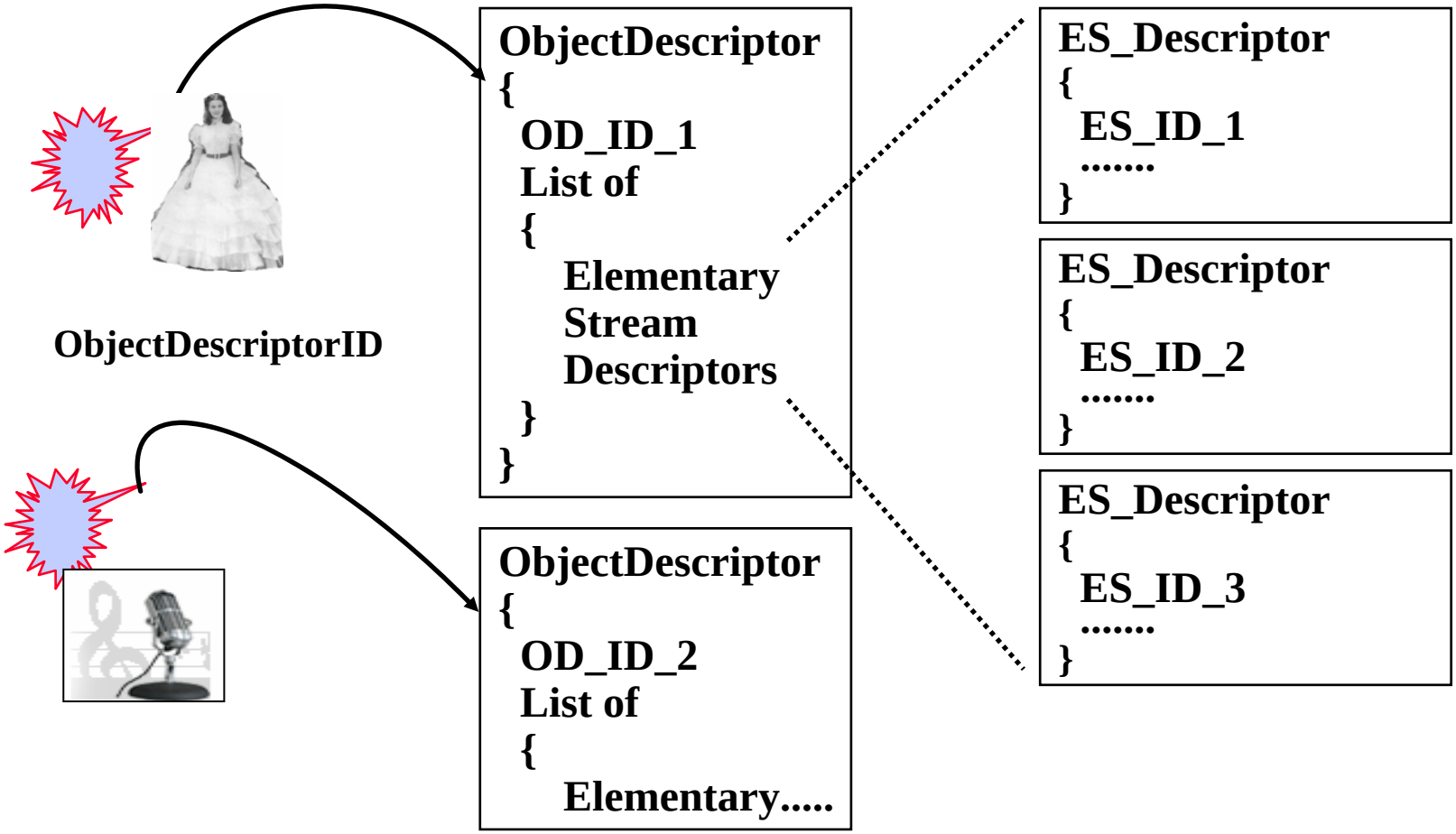
```
ObjectDescriptor
{
  OD_ID_1
  List of
  {
     Elementary.....
```

```
ObjectDescriptor
{
  OD_ID_2
  List of
  {
     Elementary
     Stream
     Descriptors
  }
}
```

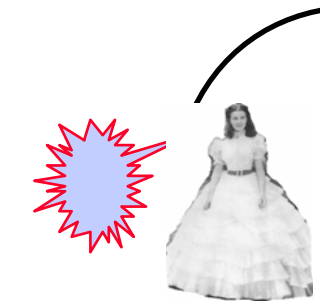
```
ES_Descriptor
{
  ES_ID_4
  .......
}
```

```
ES_Descriptor
{
  ES_ID_5
  .......
}
```

# Linking streams into the scene (4/6)



# Linking streams into the scene (5/6)



**ObjectDescriptorID**

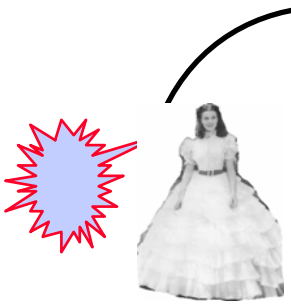
**ObjectDescriptor**
**{**
**OD_ID_1**
**URL**
***//...and nothing else!***
**}**

**ObjectDescriptor**
**{**
**OD_ID_2**
**URL**
***//...and nothing else!***
**}**

*URLs in OD point to remote ODs only*
*no local stream description needed*

*Alternative:*
*URLs to provide more flexibility*

# Linking streams into the scene (6/6)



ObjectDescriptorID

```
ObjectDescriptor
{
  OD_ID_1
  List of
  {
     Elementary
     Stream
     Descriptors
  }
}
```

```
ObjectDescriptor
{
  OD_ID_2
  List of
  {
     Elementary.....
```

```
ES_Descriptor
{
  URL
  .......
}
```

```
ES_Descriptor
{
  URL
  .......
}
```

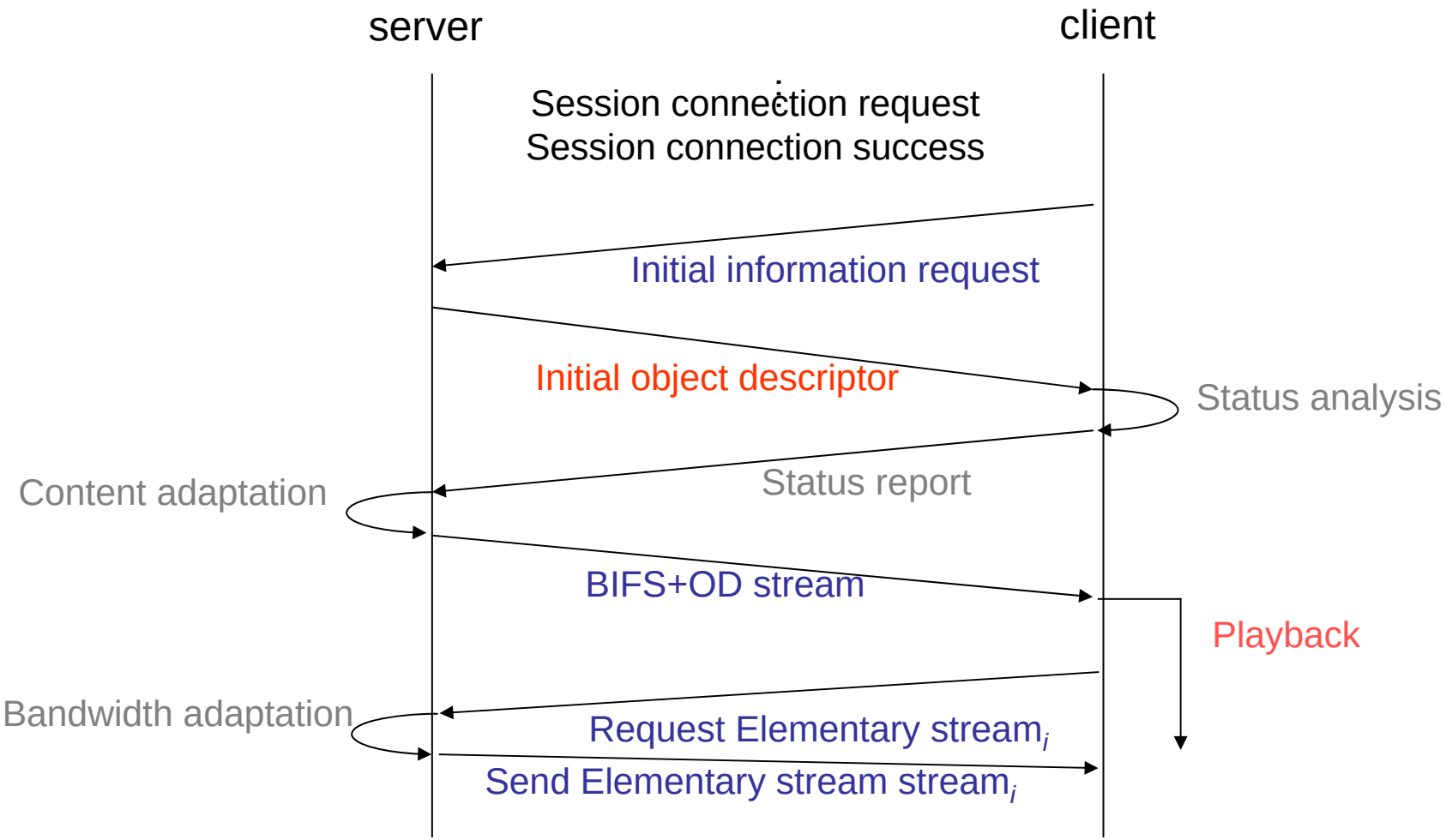
⋮ ⋮

*URLs in ESDs point to remote streams*
*All description local*

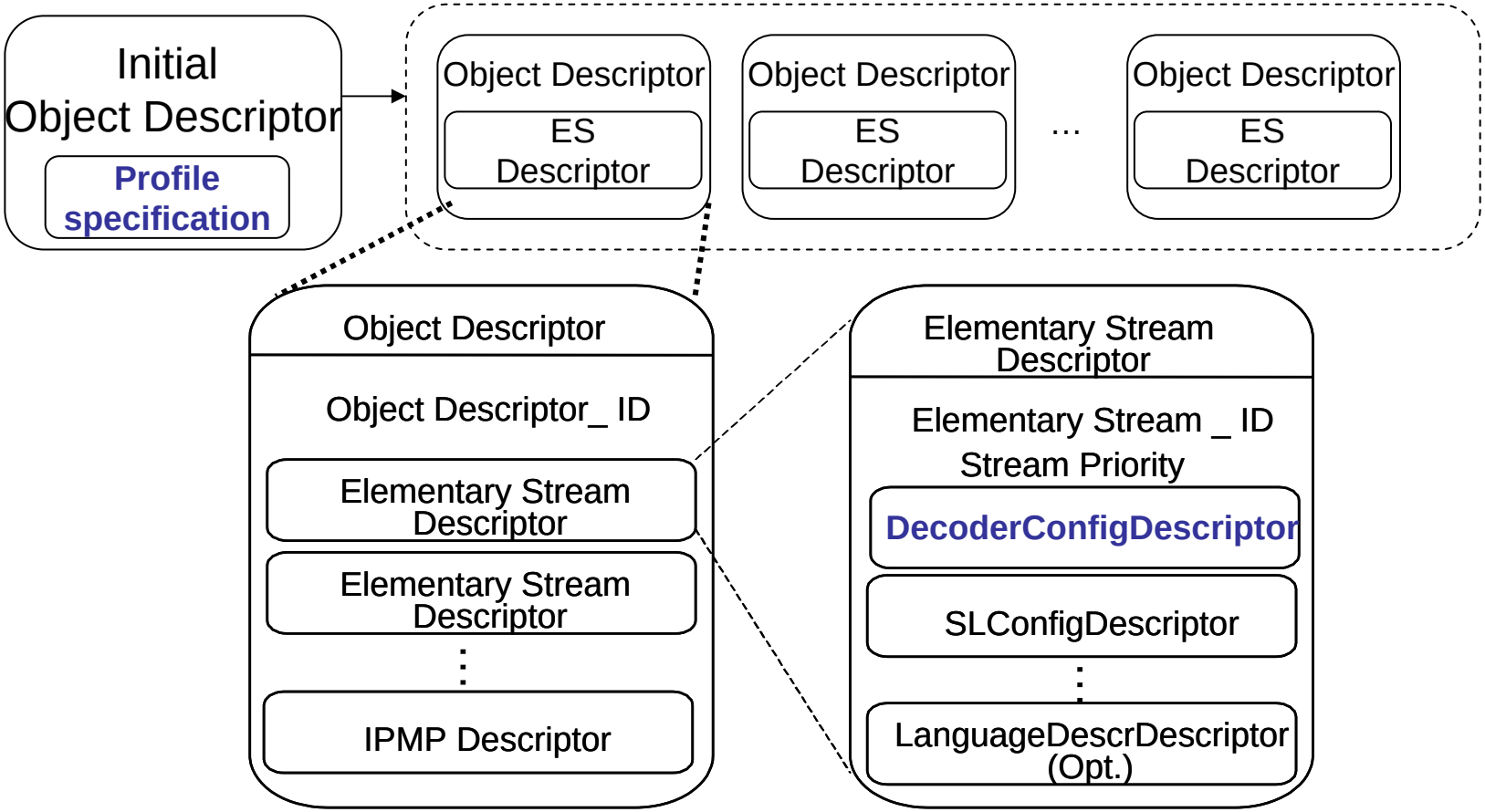
# Transport of object descriptors

- Object descriptors are encapsulated in OD commands
  - ObjectDescriptorUpdate
  - ObjectDescriptorRemove
  - ES_DescriptorUpdate
  - ES_DescriptorRemove
- OD commands are conveyed in their own ObjectDescriptorStream in a synchronized manner with time stamps
  - Objects / streams may be announced during a presentation
- Same as the scene description, there could be multiple OD streams
  - A partitioning of a large scene becomes possible
- Name scopes for identifiers (OD_ID, ES_ID) are defined
  - Resource management for sub scenes can be distributed
- A propos “resource management”
  - If the location of streams is changed, only the ODs need modification. Not the scene description

# Initial information feed-back



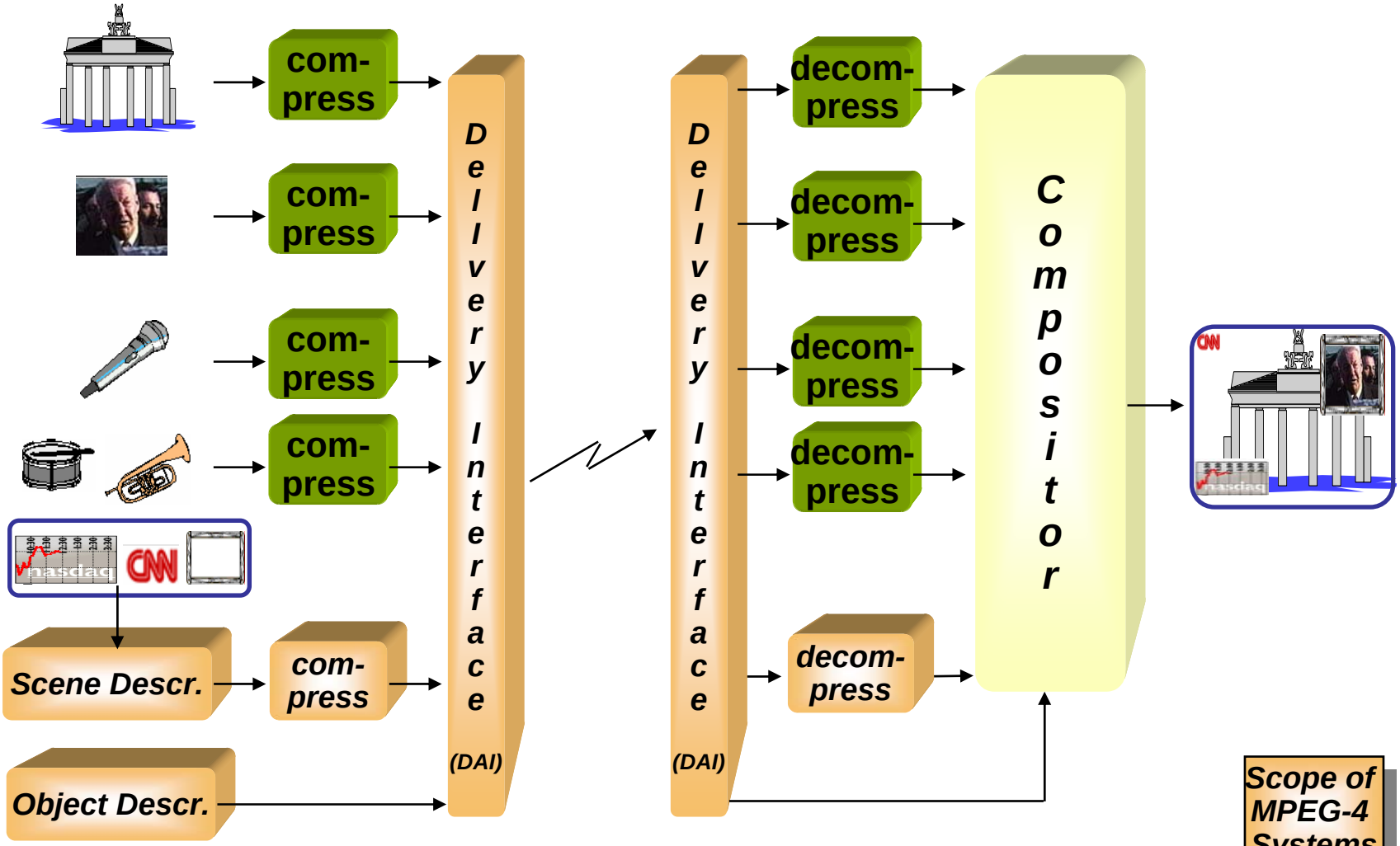
# Initial Object Descriptor(1/2)



# Initial Object Descriptor(2/2)

- Derived from the generic object descriptor
- Contains additional elements to signal profile and level
- The terminal reads the P&L indications and knows whether it has capability to process the presentation
- Profiles are signaled in multiple separate dimensions
  - Scene description
  - Graphics
  - Object descriptors
  - Audio
  - Visual
- The “first” object descriptor for an MPEG-4 presentation is always an initial object descriptor

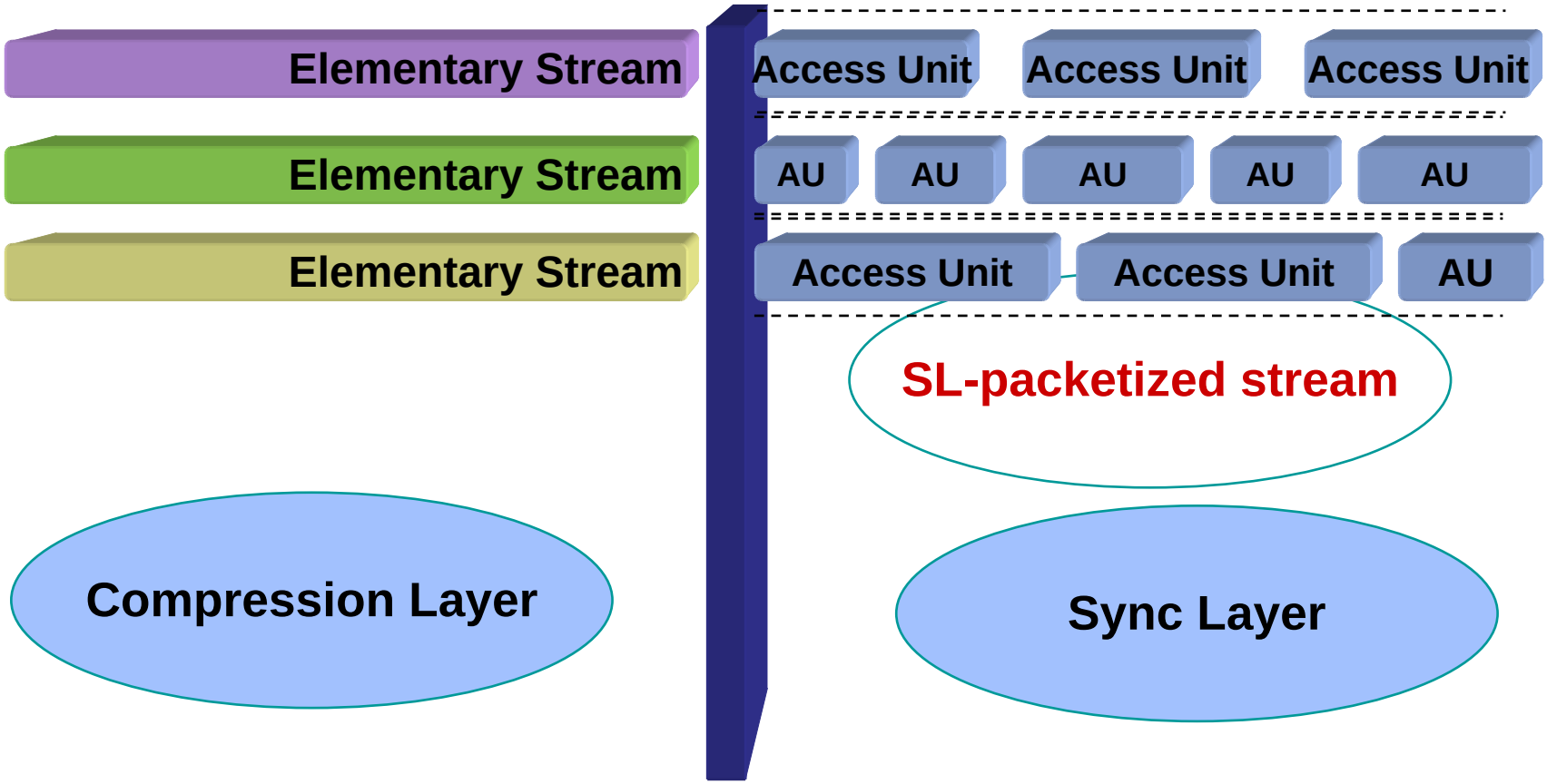
# MPEG-4 stream delivery



# Synchronization of multiple elementary streams

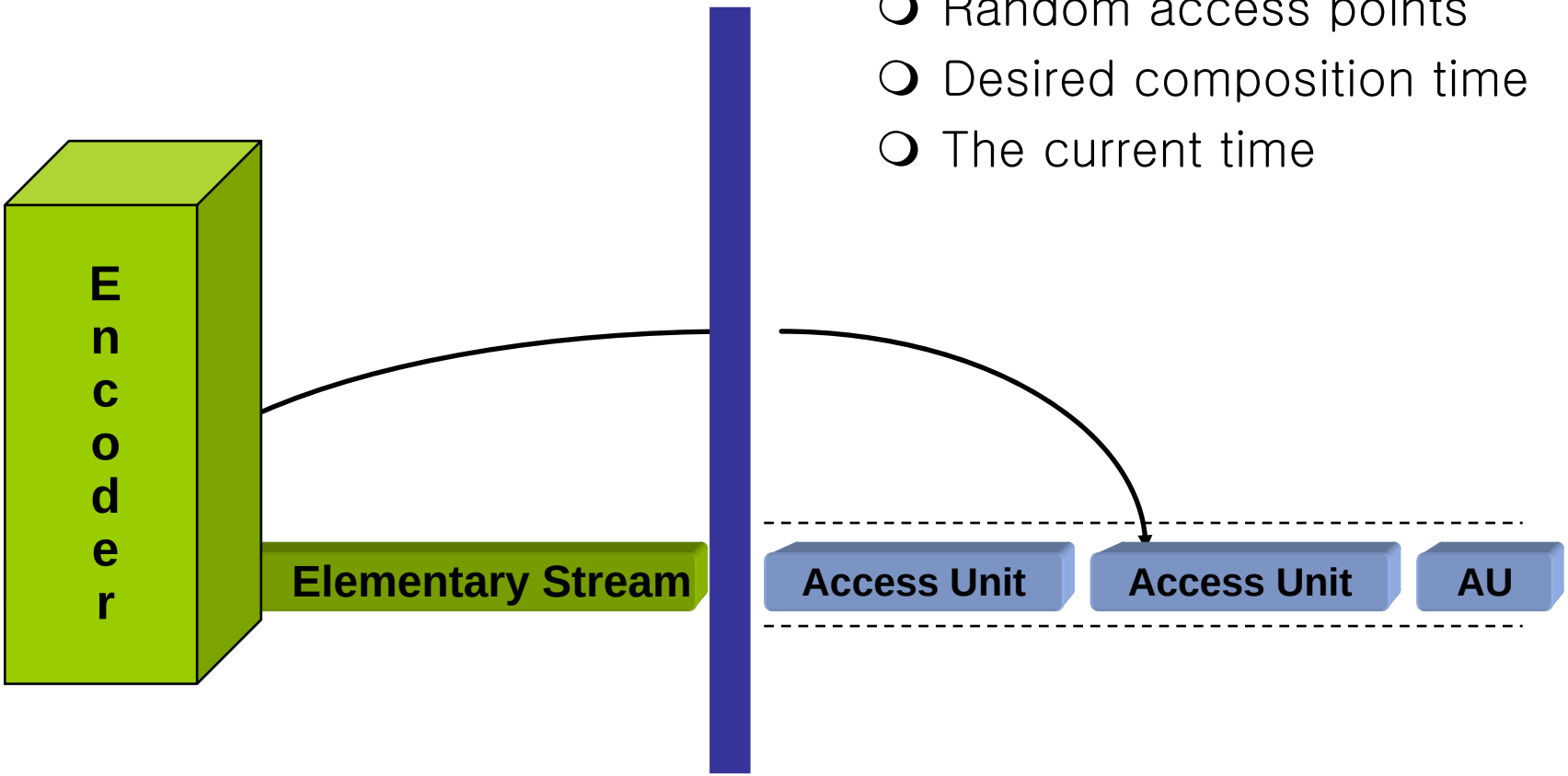
- Based on two well known concepts
- Clock references
  - Convey the speed of the encoder clock
- Time stamps
  - Convey the time at which an event should happen
- Time stamps and clock references are
  - defined in the system decoder model
  - conveyed on the sync layer

# Elementary Stream Interface



# Elementary Stream Interface

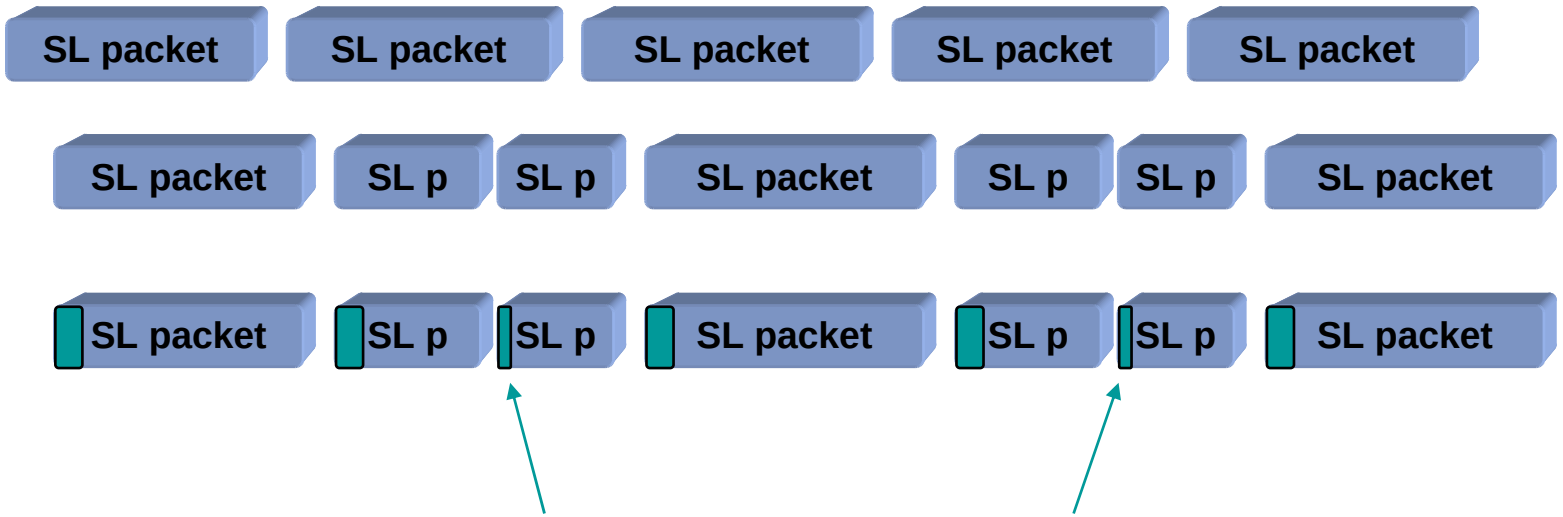
- Access unit boundaries
- Random access points
- Desired composition time
- The current time




Interface may be hidden in the Encoder

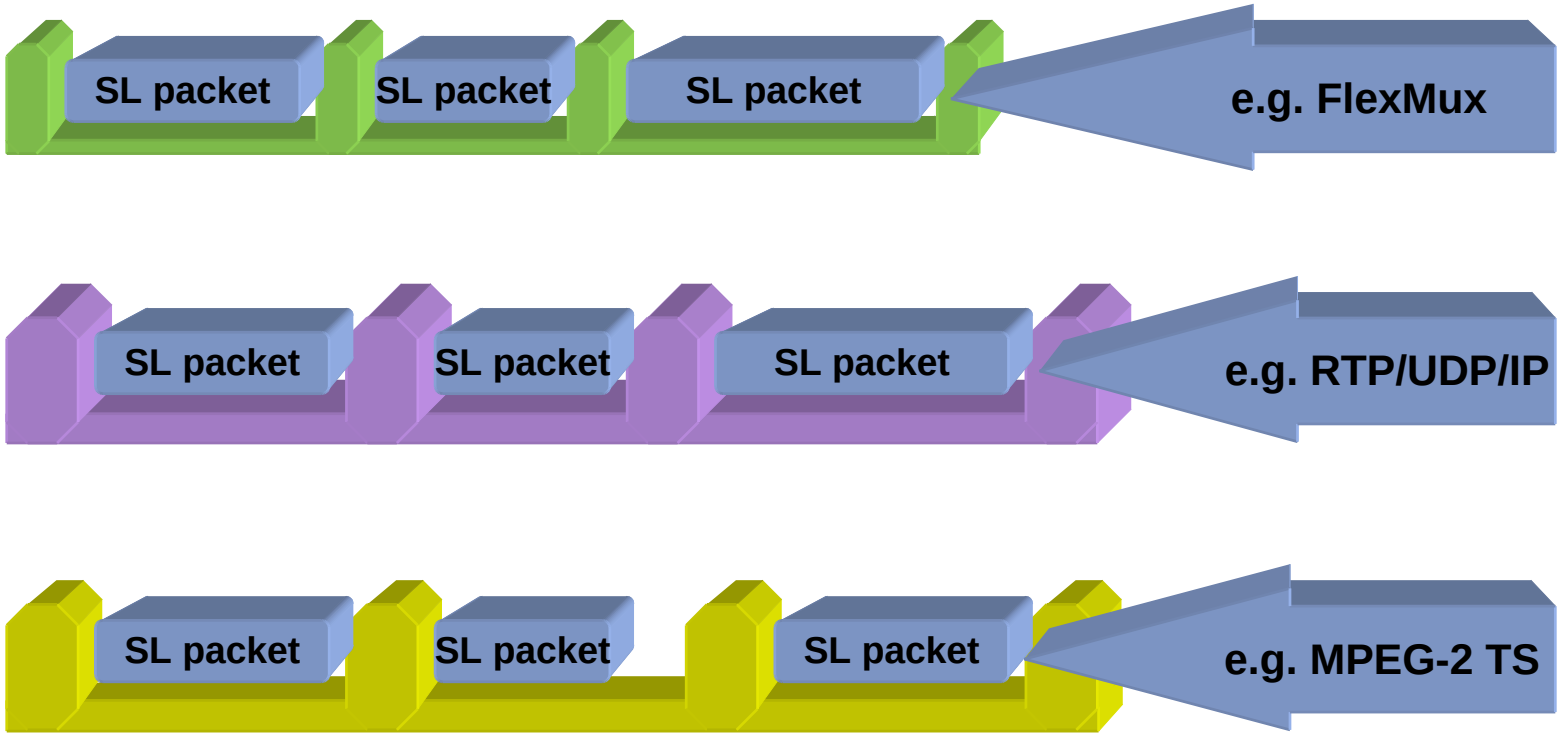
# The sync layer design

- Access units may use more than one SL packet
- Access units are conveyed in SL packets
- SL packets have a header to encode
- the information conveyed through the ESI




**SL packets that don't start an AU have a smaller header**

# Wrap SL packets in a suitable layer



# Adaptation of the sync layer syntax

- "Wrapping" of SL packets was the original design goal
- MPEG-2 TS adaptation (Amendment 7 to MPEG-2 Systems)
  and FlexMux encapsulation really work like that
- RTP encapsulation still in development
  - "Payload formats" specify the adaptation
  - They may use "Header compression", i.e.,
  - Syntax elements of SL are replaced by their RTP counterparts
    - e.g., composition time stamp --> RTP time stamp
- The MPEG-4 file format stores elementary streams
  - conversion to SL-packetized streams during playout (if needed)

# Multiplex of elementary streams

- Not a core MPEG task
- Just respond to specific needs for MPEG-4 content transmission
  - Low delay
  - Low overhead
  - Low complexity
- This prompted the design of the “FlexMux” tool
- One single file format desirable
  - This lead to the design of the MPEG-4 file format

# Modes of FlexMux

- Simple Mode **(if Index < 240)**

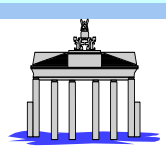
Index = FlexMux Channel number
Length = Length of payload in byte

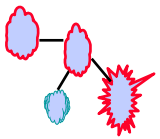

**FlexMux packet header**

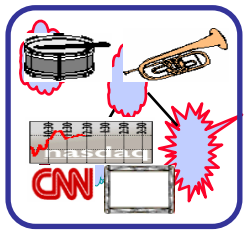
MuxCode Mode **(if Index ≥ 240)**

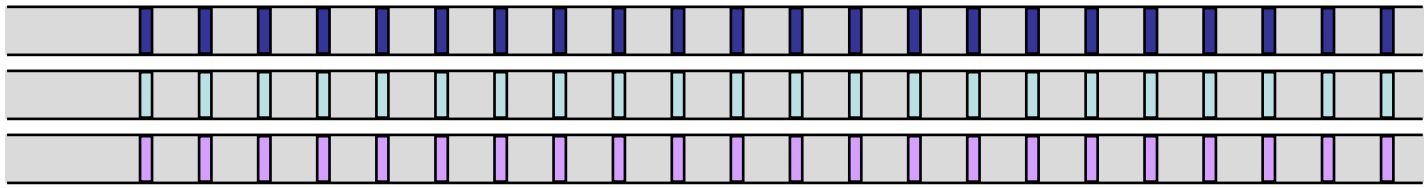
Index = points to a MuxCodeTableEntry

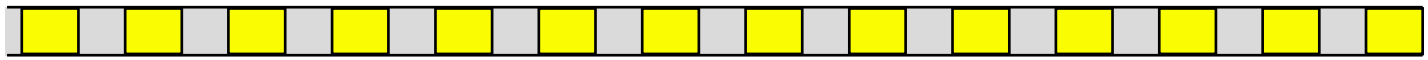
# A multiplex example

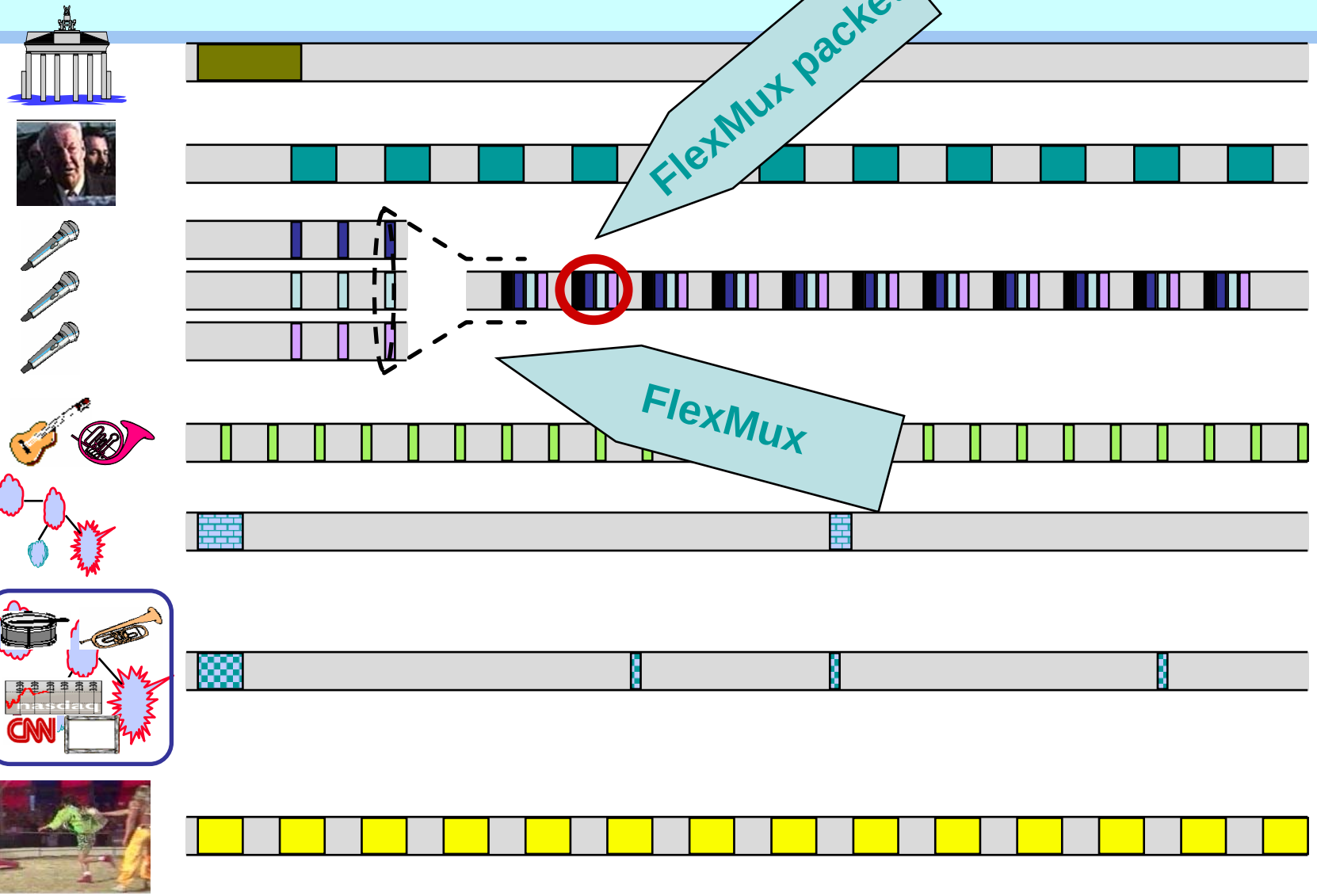


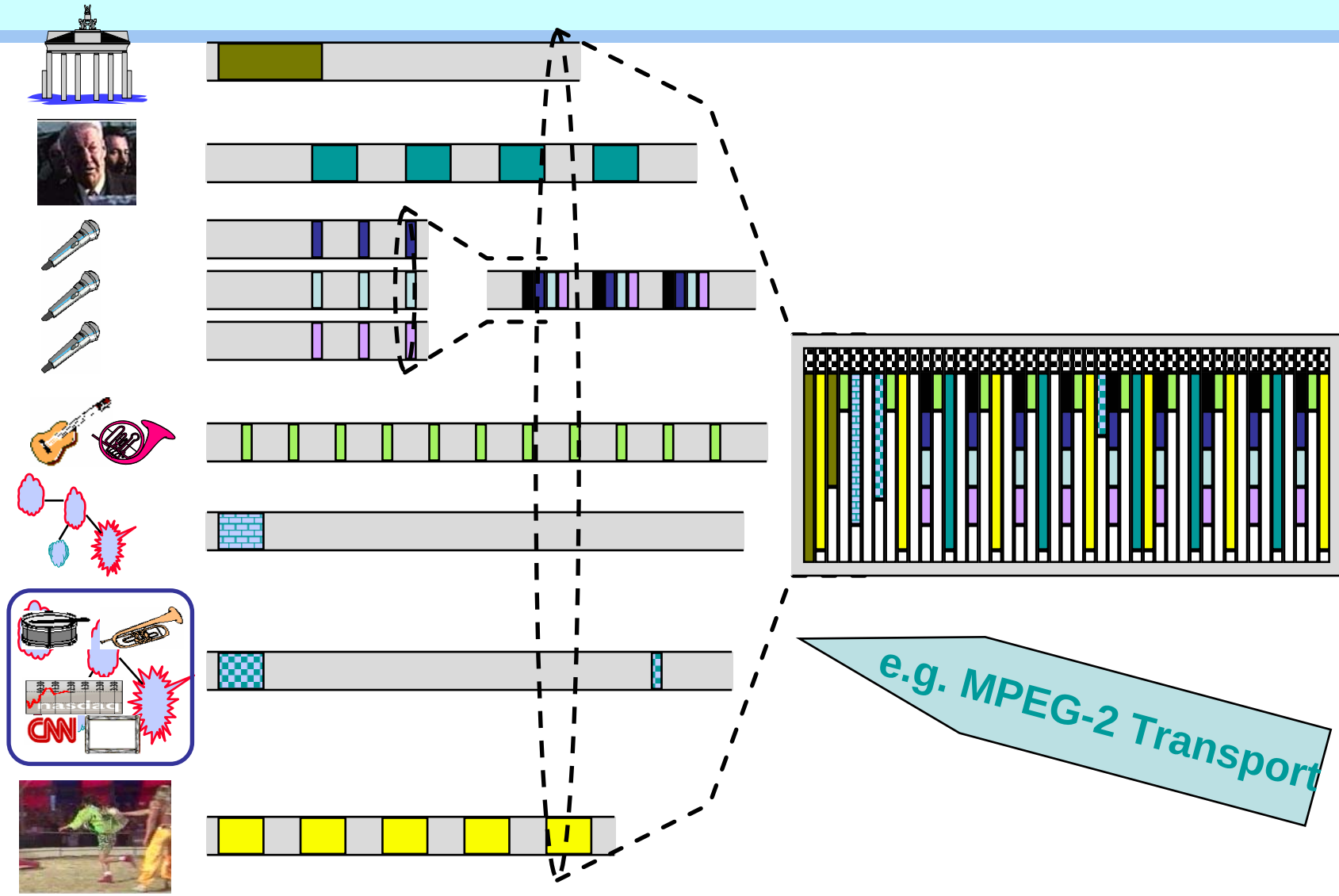







FlexMux packet
FlexMux
CNN


e.g. MPEG-2 Transport

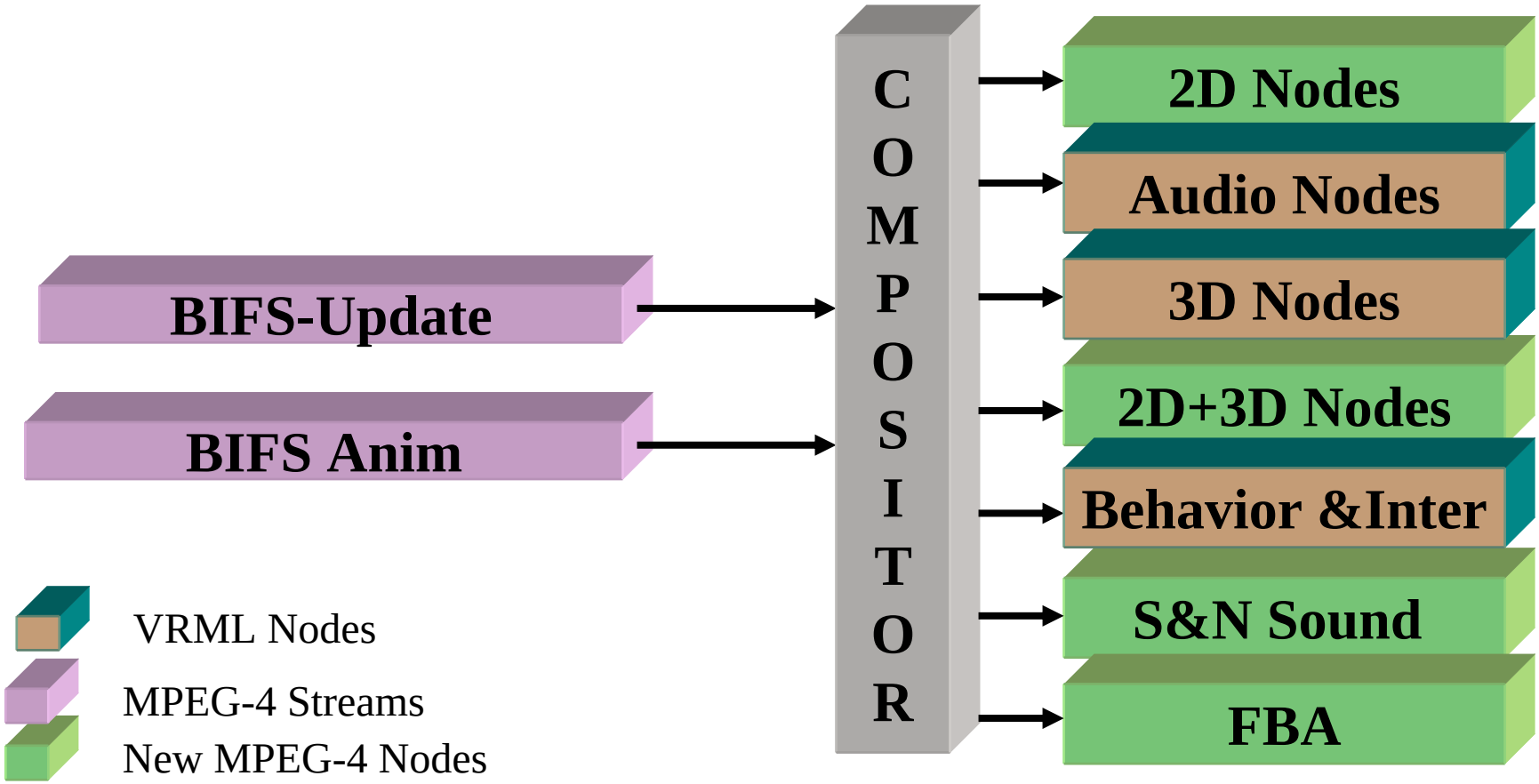
# BIFS (Binary Format for Scene)

- A Scene description for composition is needed because MPEG-4 is object-based representation system

- BIFS is the MPEG-4 Scene description protocol
  - to compose MPEG-4 objects (temporal & spatial)
  - to describe interaction with MPEG-4 objects
  - to animate MPEG-4 objects

- BIFS is based on VRML

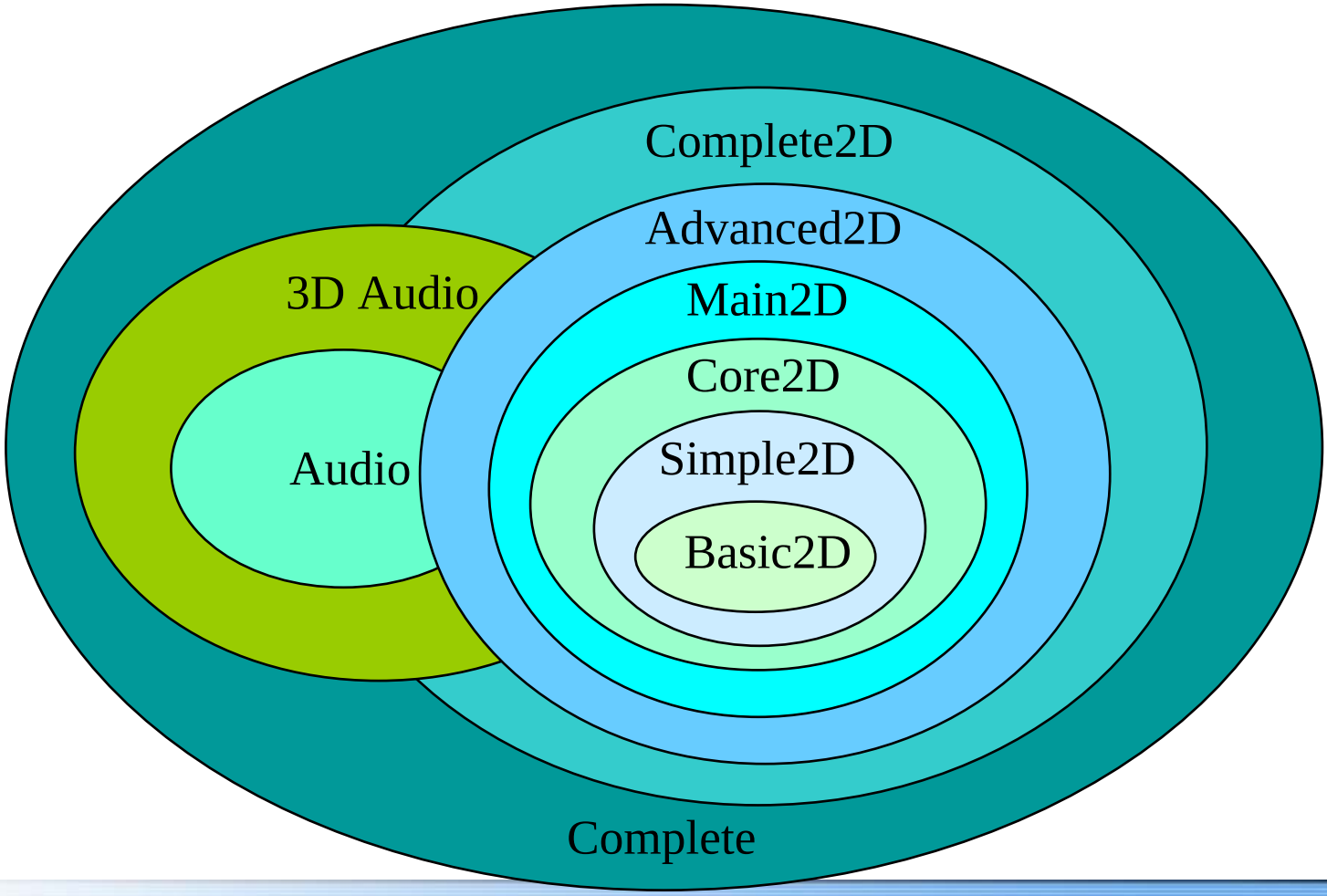
# BIFS Components



# BIFS Scene Features

- Audio video (objects) playback
- 2D Composition & Graphics
- 3D Composition & Graphics (VRML inclusive)
- Advanced audio composition
- Interactivity and Behavior
- Scripting (javascript)
- Face Animation
- Body Animation

# Scene Graph Profile



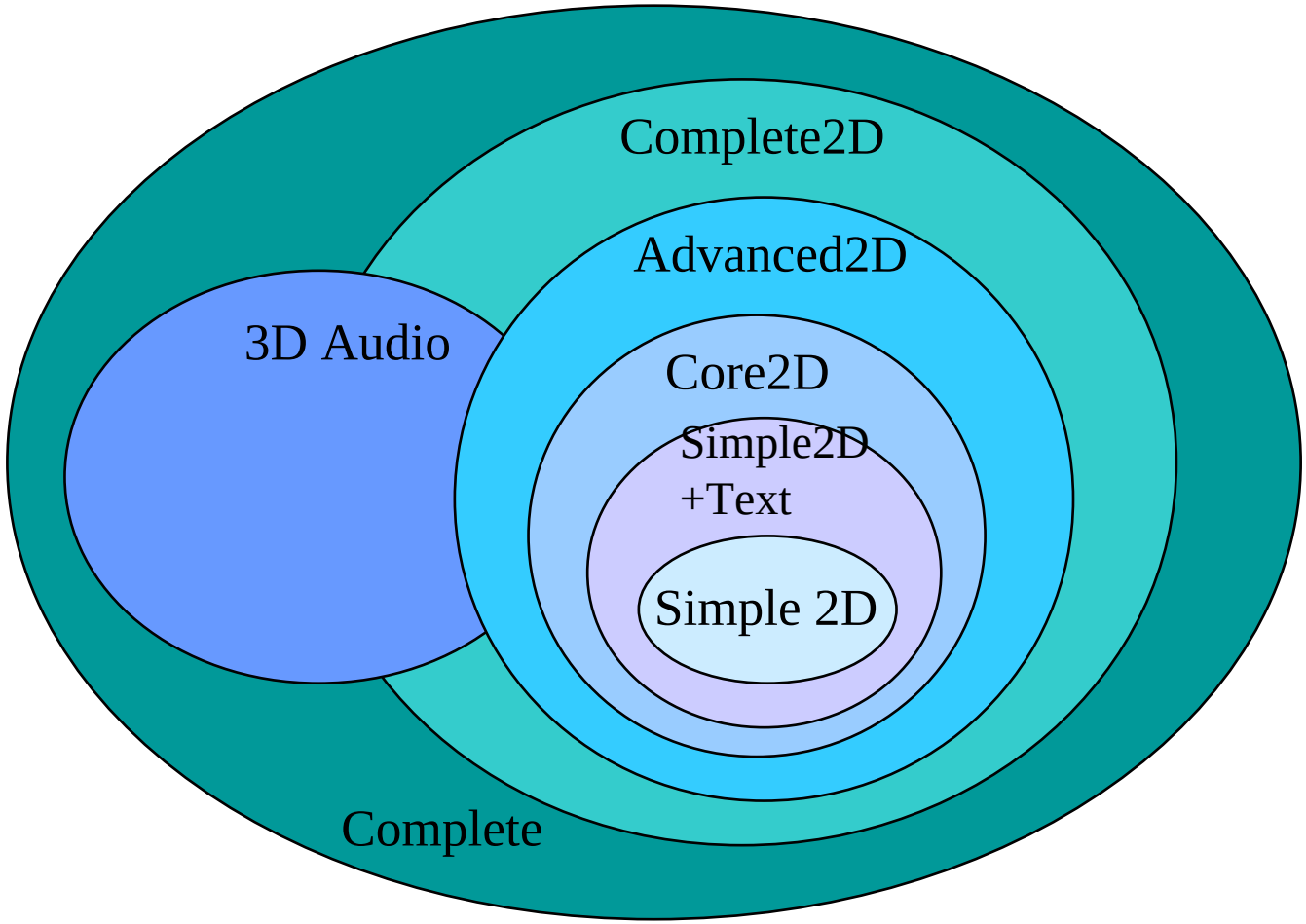
# Scene Graph Profile

- Basic2D
  - audio-video only
- Simple2D
  - Basic2D + scene update
  - like broadcast TV. No interaction
- Core2D
  - 2D composition, 2D texturing
  - local interaction, local animation
  - BIFS-updates, quantization,
  - web links, sub-schenes
  - back channel (ServerCommand) and VoD features (MediaControl, MediaSensor)

# Scene Graph Profile

- Main2D
  - Core2D capability
  - Flextime node for advanced, flexible synchronization
- Advanced2D
  - comprise all Basic2D and Core2D Scene graph functionality
  - advanced 2D composition, advanced local interaction, BIFS-Anim, scripting, advanced audio
- Complete2D
  - all 2D scene description elements of the BIFS tool
  - 2D transformation and alpha blending

# Graphics Profile



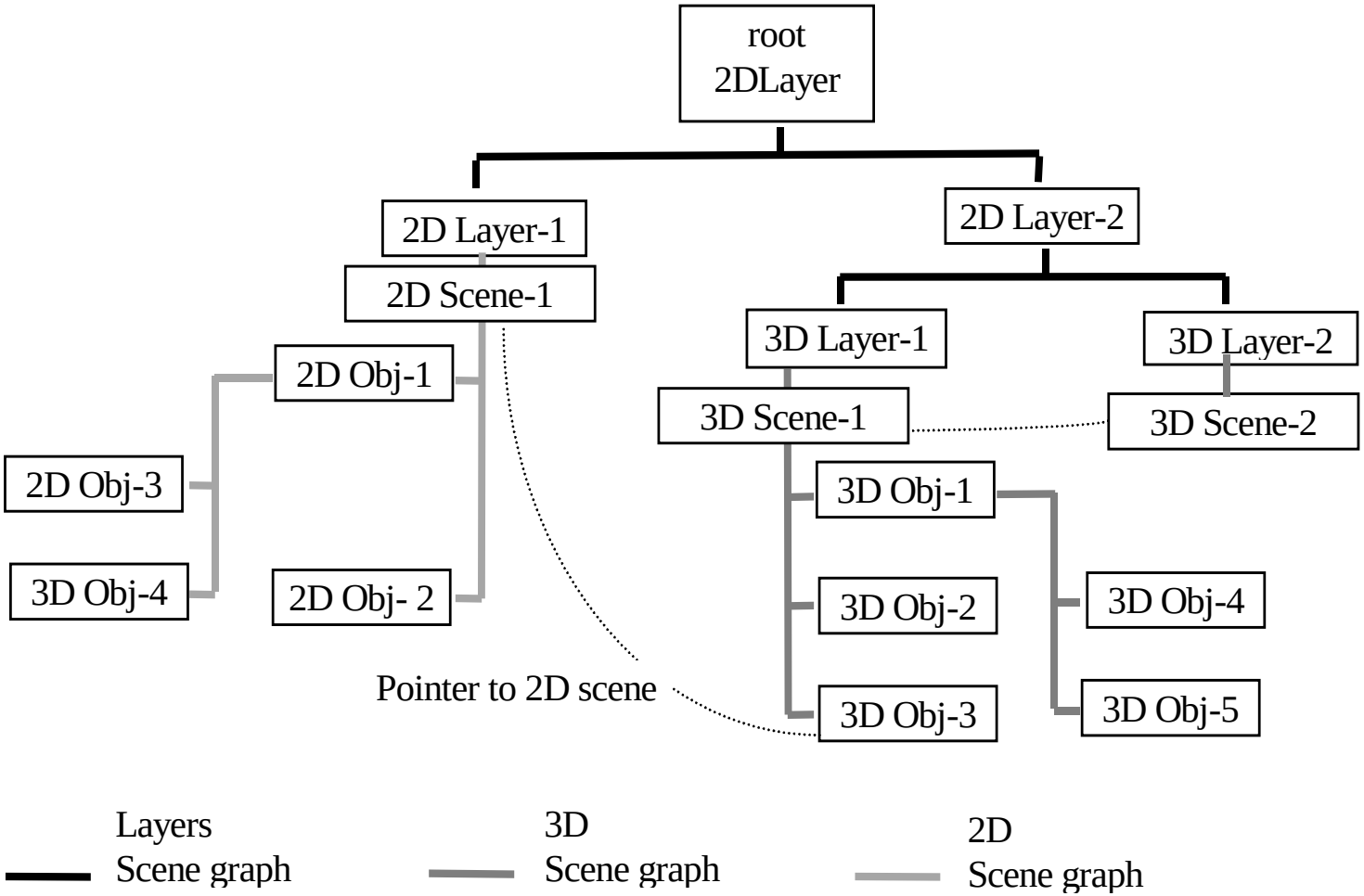
# Graphics Profile

- Simple2D
  - to support Scene Graph Basic2D profile
- Simple2D+Text
  - supports Text
- Core2D
  - designed for applications using some simple graphic elements in addition to audio and visual objects
- Advanced2D
  - designed for applications using advanced graphics elements in addition to audio and visual objects
- Complete2D
  - provides two-dimensional graphics functionalities
  - supports features such as arbitrary two-dimensional graphics and text in conjunction with visual objects

# Interactivity and Behaviors

- VRML Based event model
  - Sensors generate events according to some user or scene events.
  - Interpolator enable to generate time varying attributes for animation.
  - ROUTEs enable to pass events between nodes
- BIFS Update Protocol
  - Used to modify of the Scene.
  - Supported commands :
    - Replace Scene
    - Add/Remove object
    - Change Scene Properties
    - Add/Remove Behaviors

# Scene Graph Example



# Empty Initial Scene

```
InitialObjectDescriptor {
        objectDescriptorID 1
        ODProfileLevelIndication 254
        sceneProfileLevelIndication 254
        audioProfileLevelIndication 254
        visualProfileLevelIndication 254
        graphicsProfileLevelIndication 254

        esdescr [
                ES_Descriptor {
                        es_id 1
                                decConfigDescr DecoderConfigDescriptor {
                                                streamType 3
                                                objectTypeIndication 1
                                                decSpecificInfo BIFSConfig {
                                                                isCommandStream true
                                                                pixelMetrics        true
                                                                pixelWidth          100
                                                                pixelHeight         100
                                                }
                                  }
                                slConfigDescr SLConfigDescriptor {}
                                }
                ]
```

| streamType value | stream type description |
|---|---|
| 0x00 | Forbidden |
| 0x01 | ObjectDescriptorStream |
| 0x02 | ClockReferenceStream |
| **0x03** | **SceneDescriptionStream** |
| **0x04** | **VisualStream** |
| **0x05** | **AudioStream** |
| 0x06 | MPEG7Stream |
| 0x07 | IPMPStream |
| 0x08 | ObjectContentInfoStream |
| 0x09 | MPEGJStream |
| 0x0A - 0x1F | reserved for ISO use |
| 0x20 - 0x3F | user private |

# 2D graphical context

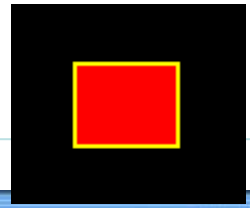
- Filling and striking
  - Appearance

| | | |
|---|---|---|
| exposedField | SFNode | **material** |
| exposedField | SFNode | **texture** |
| exposedField | SFNode | **textureTransform** |

  - Material2D

| | | |
|---|---|---|
| exposedField | SFColor | **emissiveColor** |
| exposedField | SFBool | **filled** |
| exposedField | SFNode | **lineProps** |
| exposedField | SFFloat | **transparency** |

```
OrderedGroup {
    children [
        Shape {
            appearance Appearance {
                material Material2D {
                        emissiveColor 1.0 0.0 0.0
                        filled true
                        lineProps LineProperties {
                                lineColor 1.0 1.0 0
                                width 2.0
                        }
                }
            }
            geometry Rectangle {
                        size 50.0 40.0
            }
        }
    ]
}
```



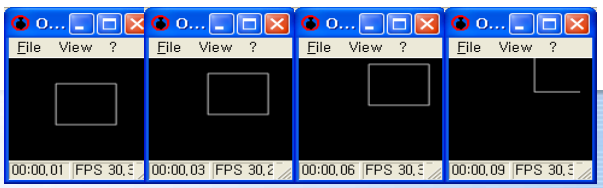
# A complete animated scene

– TimeSensor

| | | |
|---|---|---|
| exposedField | SFTime | **cycleInterval** |
| exposedField | SFBool | **enabled** |
| exposedField | SFBool | **loop** |
| exposedField | SFTime | **startTime** |
| exposedField | SFTime | **stopTime** |
| eventOut | SFTime | **cycleTime** |
| eventOut | SFFloat | **fraction_changed** |
| eventOut | SFBool | **isActive** |
| eventOut | SFTime | **time** |

•PositionInterpolator2D

| | | |
|---|---|---|
| eventIn | SFFloat | **set_fraction** |
| exposedField | MFFloat | **key** |
| exposedField | MFVec3f | **keyValue** |
| eventOut | SFVec3f | **value_changed** |




```
OrderedGroup {
  children [
    DEF N0 Transform2D {
      children [
    Shape {
      geometry Rectangle {
        size 50.0 40.0
      }
    }
      ]
    }
    DEF N2 TimeSensor {
      cycleInterval 10.0
    }
    DEF N1 PositionInterpolator2D {
      key [ 0.0 1.0 ]
      keyValue [ 0.0 0.0 40.0 40.0 ]
    }
  ]
}
ROUTE N2.fraction_changed to N1.set_fraction
ROUTE N1.value_changed to N0.translation
```

# BIFS command

- Replace

```
<Replace atNode="Identifier">
... the new node to replace
with ... </Replace>
```

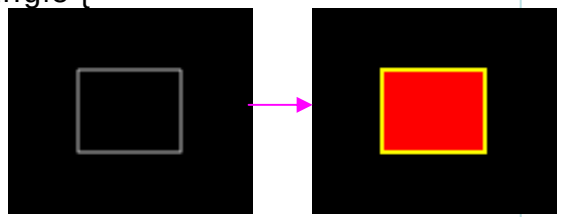
- Insert

```
<Replace atNode="identifier"
atField="Fieldname"
position="FIRST|END|n">
... new node ...
</Replace>
```

- Delete

```
<Delete atNode="identifier"
atField="FieldName"
position="FIRST|END|n"/>
```

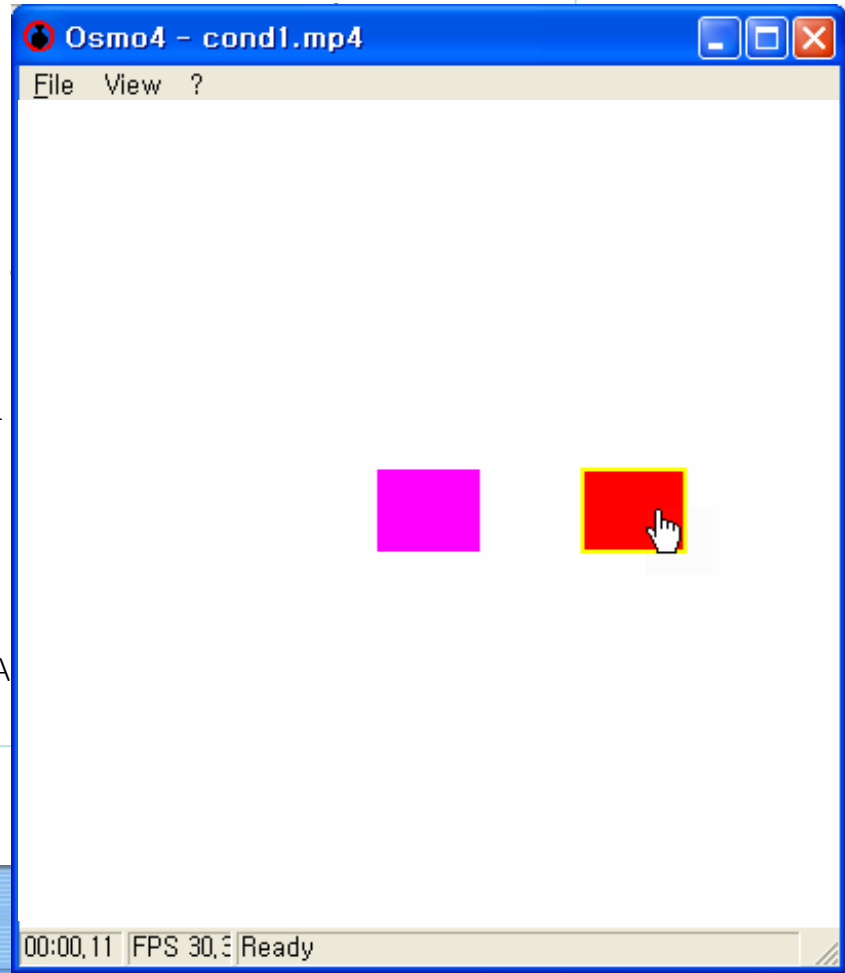
```
OrderedGroup {
 children [
  DEF N0 Shape {
   geometry Rectangle {
    size 50.0 40.0
   }
  }
 ]
}
AT 2000 {
REPLACE N0 BY Shape {
 appearance Appearance {
  material Material2D {
   emissiveColor 1.0 0.0 0.0
   filled true
   lineProps LineProperties {
    lineColor 1.0 1.0 0.0
    width 2.0
   }
  }
 }
 geometry Rectangle {
  size 50.0 40.0
 }
}
}
```



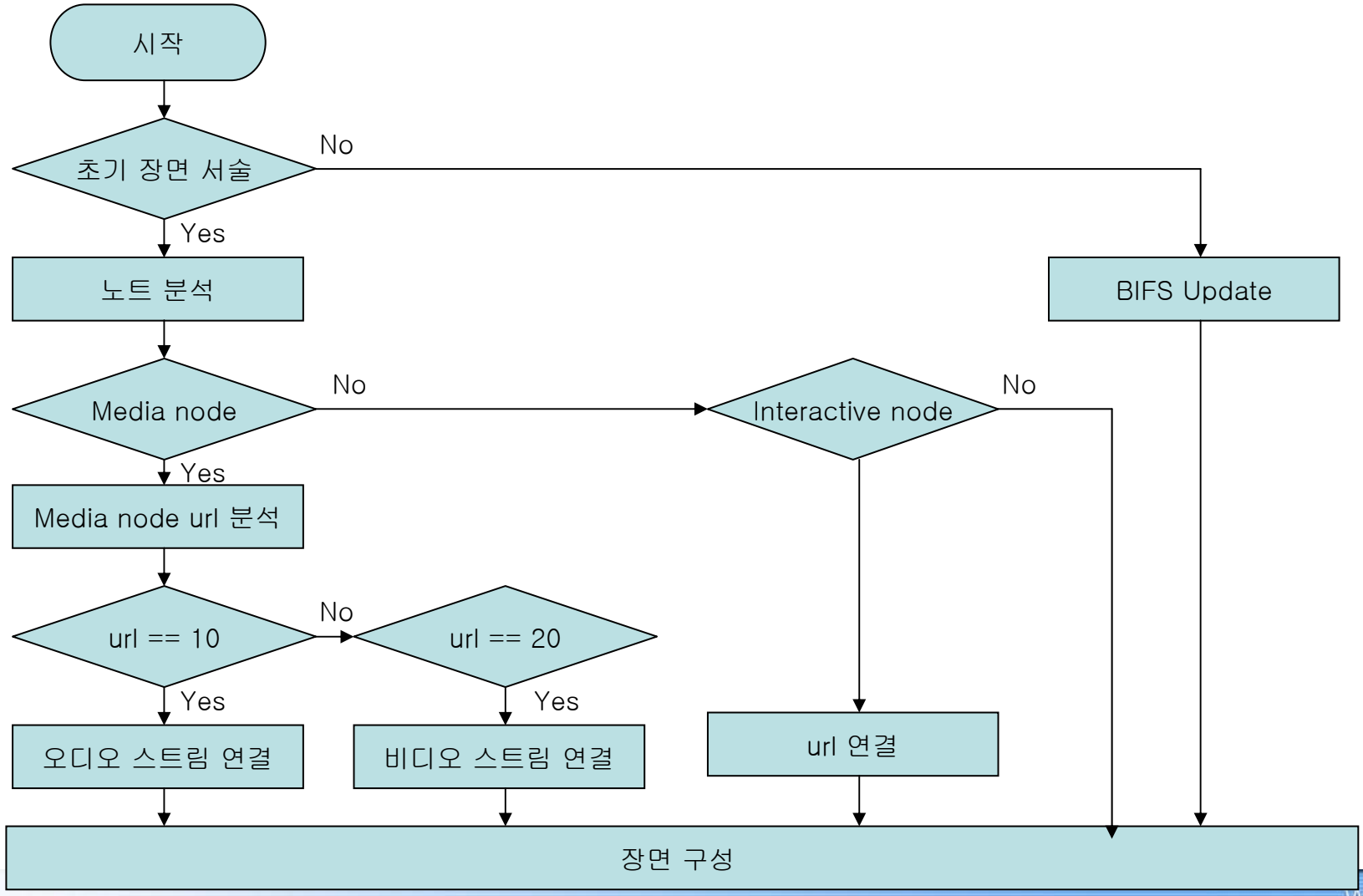
# Interactivity

```
OrderedGroup {
  children [
    Background2D {
      backColor 1.0 1.0 1.0
    }
    Transform2D {
      translation 100.0 0.0
      children [
                DEF N1 TouchSensor {
                }
                Shape {
                  appearance Appearance {
                    material Material2D {
                      emissiveColor 1.0 0.0 0.0
                      filled true
                      lineProps LineProperties {
                                lineColor 1.0 1.0 0.0
                                width 2.0
                      }
                    }
                  }
                  geometry Rectangle {
                    size 50.0 40.0
                  }
                }
      ]
    }
```

```
DEF N2 Transform2D {
    }
    DEF N0 Conditional {
      buffer {
                INSERT AT N2.children [0] Shape {
                  appearance Appearance {
                }
      }
    }
  ]
}
ROUTE N1.isA
```

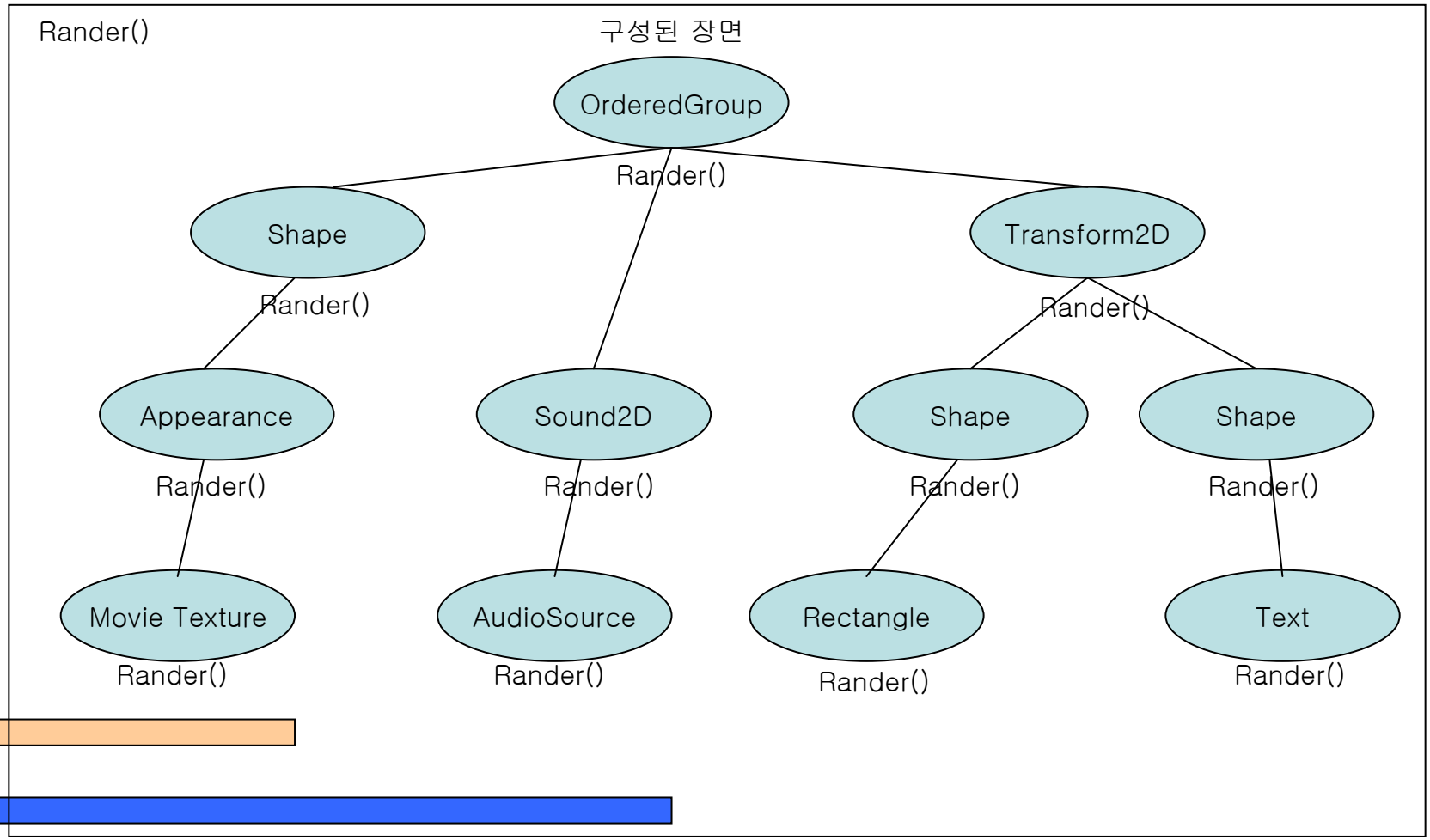


# BIFS 복호화



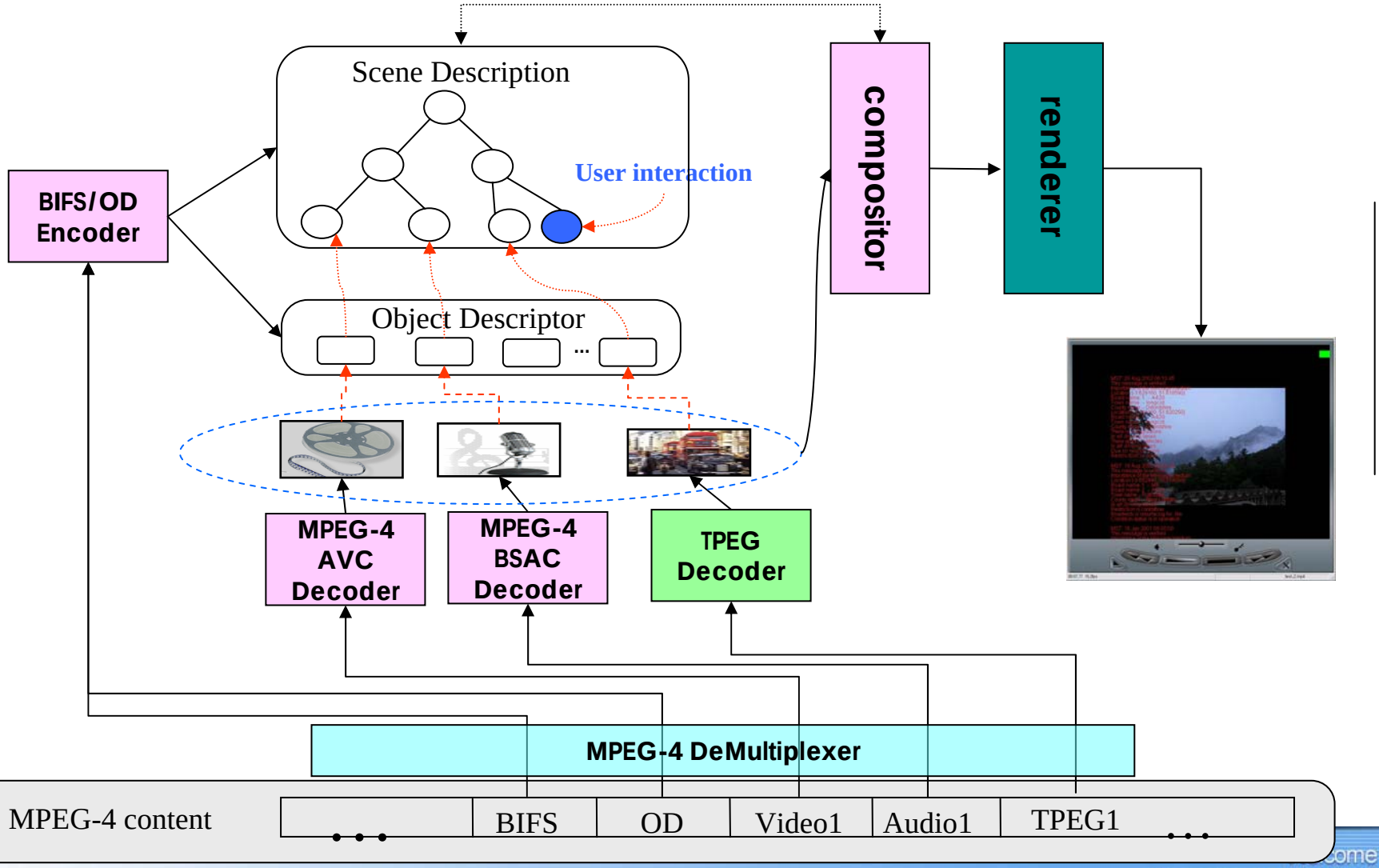
# 프리젠터

Presenter {




}

# MPEG-4 Content Presentation



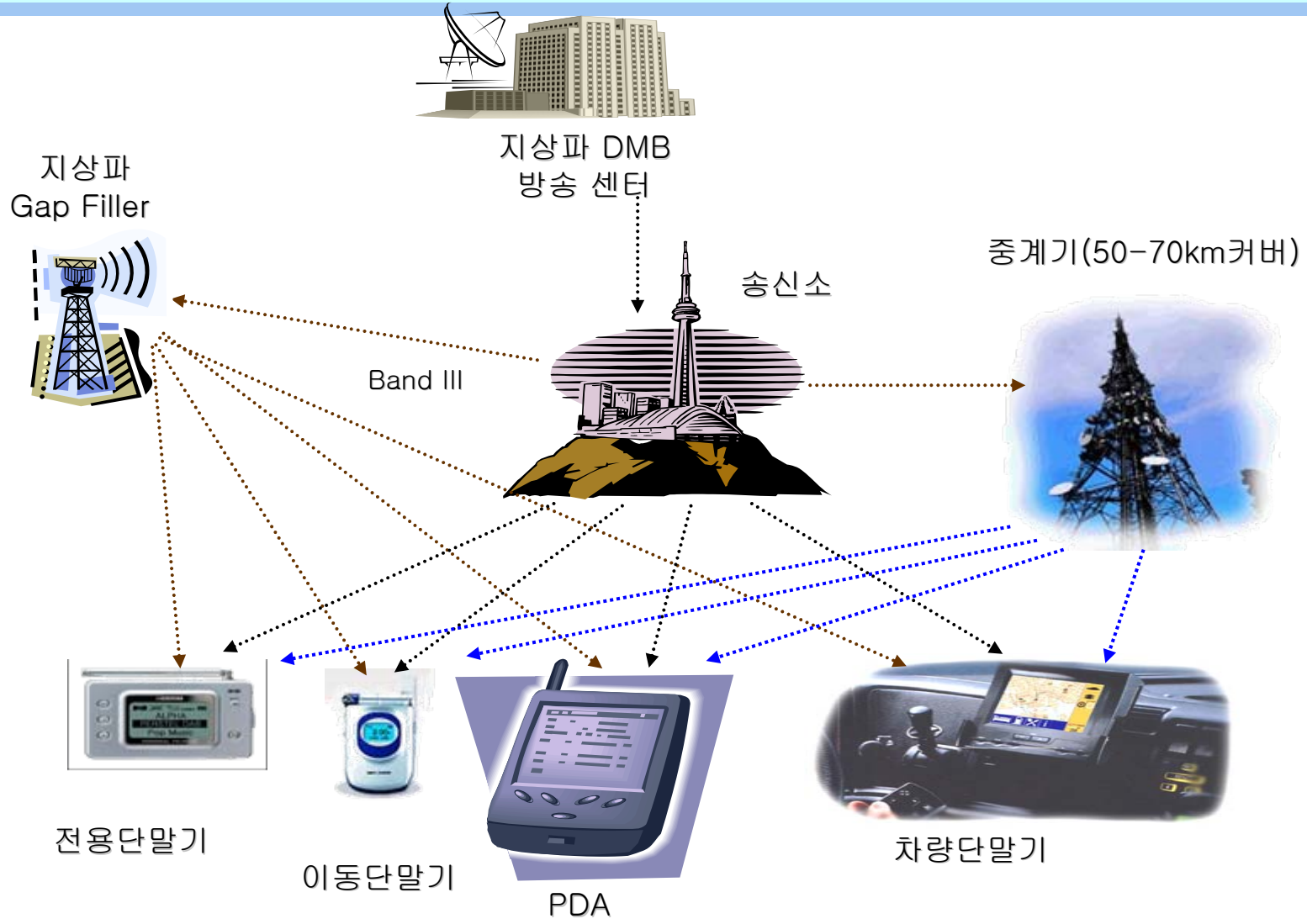
# 이동 멀티미디어 서비스

# 이동 멀티미디어 방송 개요

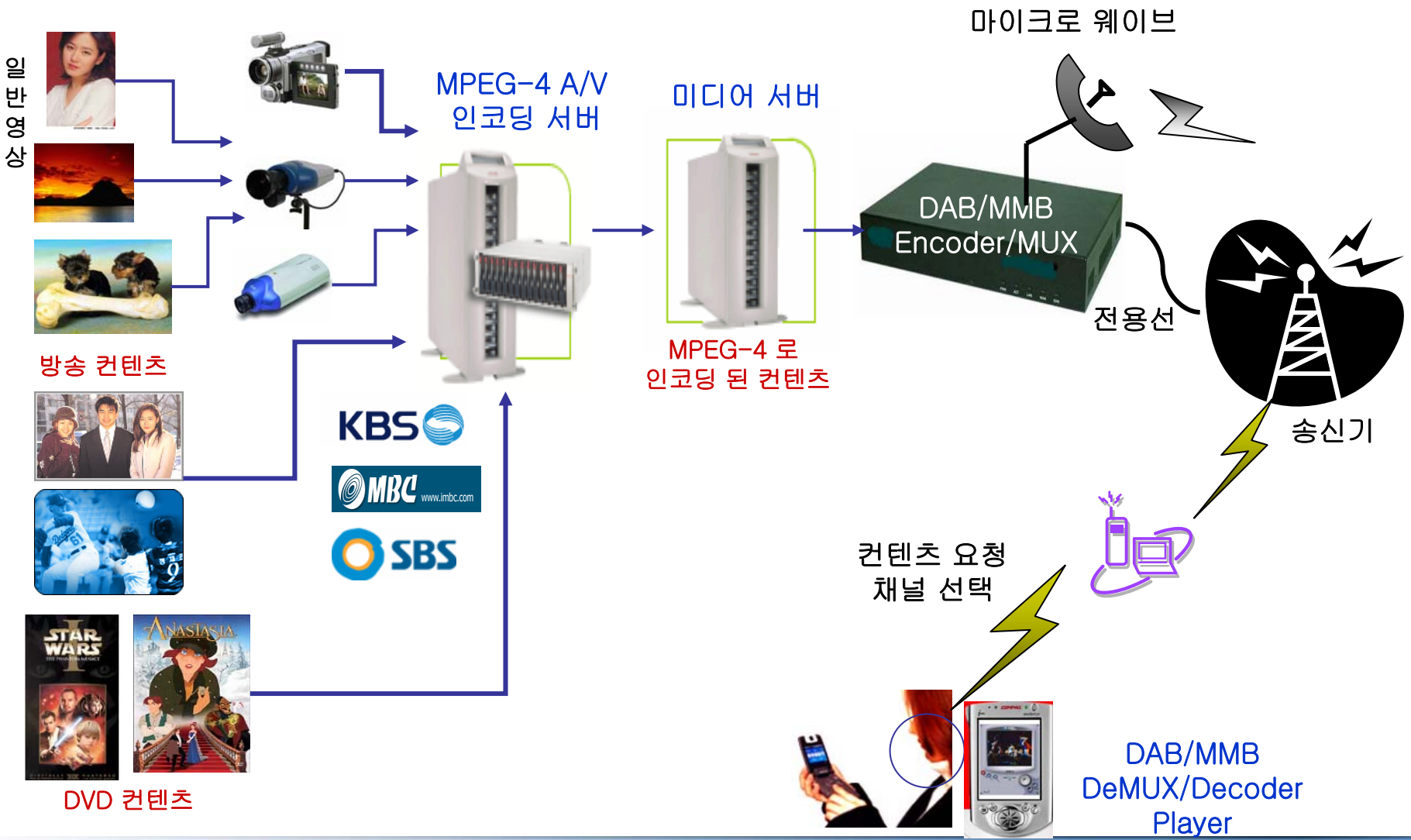
- 고품질의 음성 및 영상 서비스를 언제 어디서나 제공할 수 있는 이동멀티미디어
  - '듣는 방송'에서 '보고 듣는 방송'으로 라디오방송의 개념을 확장
  - 뛰어난 이동수신 특성을 바탕으로 음악·문자·동영상 등 다양한 컨텐츠를  소형TV·PDA 등 휴대용 단말을 통해 전달
  - 고화질·고음질을 추구하는 디지털 지상파TV방송과 보완적인 관계 구축
  - PDA, 이동통신단말기와 결합된 서비스 제공을 통해 이동단말기의 획기적인 수요창출

- 지상파 디지털 멀티미디어 방송(지상파 DMB)
  - MPEG-4에 기반한 이동 방송과 멀티미디어 서비스
- 위성 디지털 멀티미디어 방송(S-DMB)

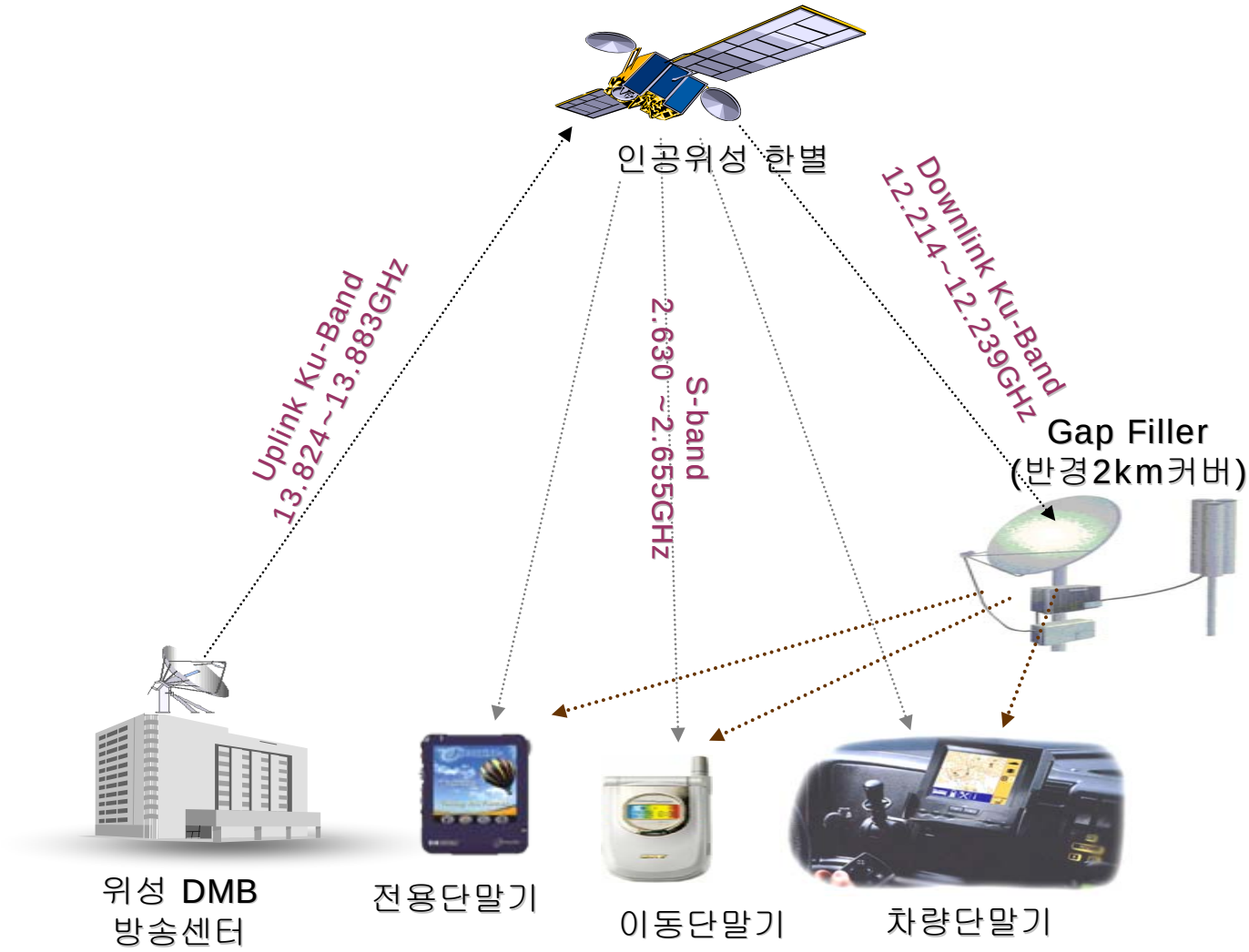
# 지상파 DMB 서비스



# 지상파 DMB 컨텐츠 전송 개념



# 위성 DMB



# 위성 DMB

- 서비스 타입
  - 오디오 : CD Quality Stereo Audio
  - 멀티미디어 서비스
    - Video : 최대 352 X 240 @15fps
    - Audio : Above FM Quality(Max. CD)
    - Data : Text and Graphic
  - Data Service
    - EPG, Interactive Services, Etc.
  - 속도 150km/h에서도 수신 가능할 것
- 표준
  - 오디오 : MPEG-2 AAC + SBR
  - 비디오 : H.264 Baseline Profile Level1.3
  - 다중화 : ISO/IEC 13818-1(MPEG-2)
  - PSI/SI : ISO/IEC 13818-1및 EN300 468
  - DVB-SI
  - 다운로드 : ISO/IEC 13818-6(DSM-CC)
  - Data Carousel

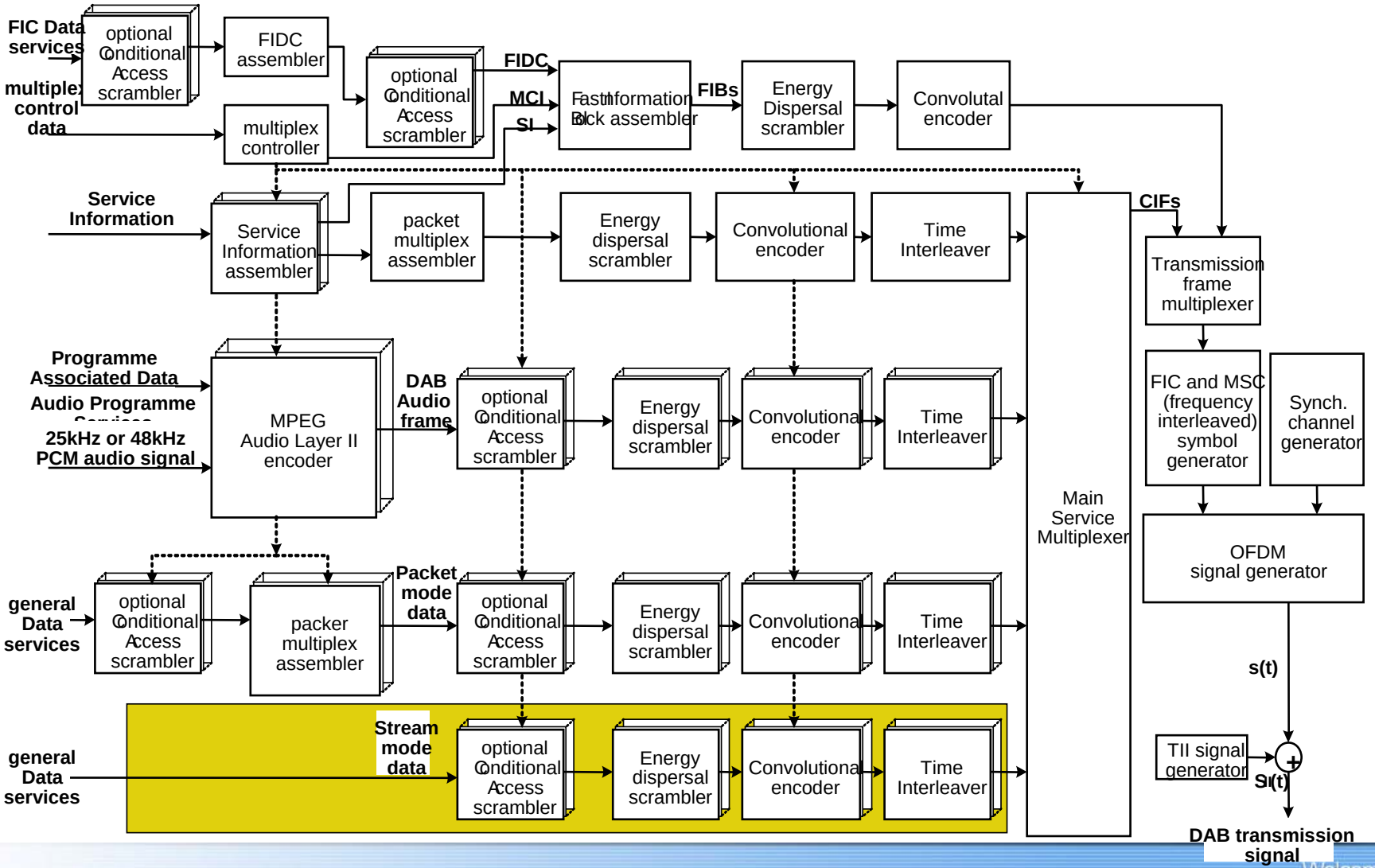
# DMB Technology Trend

- Service
  - DAB(Digital Audio Broadcasting) service : Europe, Canada…
  - DVB-T based Mobile TV service : Singapore
  - ISDB-T based DMB service : Japan
  - Korea
    - Audio only service -> Mobile TV service
    - Optimal standard -> Eureka 147 + MPEG-4 AV
- Terminal
  - Japan
    - Satellite DMB receiver : Toshiba
    - Terrestrial DMB cellular-phone prototype
  - Korea
    - Commercial DMB terminal/ Main SoC development

# Eureka-147

- ’87년 Eureka-147 프로젝트 결성
- ’92년 새로운 주파수대를 사용하는 시스템 개발, 시험 실시
- ’95년 영국의 BBC에서 시험방송 돌입, 민간 방송사업자에게는 1998년 10월에 Digital One사에 전국 면허를 부여하여 1999년 10월부터 방송 개시
- 유럽의 몇몇 나라가 ’96년 후반에서 ’97년 중반 사이에 DAB 상용 서비스 시작
- ’01년 공청회를 통하여 국내 지상파 디지털 오디오 방송 잠정 표준안으로 결정
- ’02년 실험 전담반을 구성하여 실험 방송 실시
- ’02년 국내 지상파 디지털 오디오 방송 표준안으로 결정

# Eureka-147 Structure



# 지상파 DMB 서비스 요구사항

- 지상파 DBM 방송센터
  - Ch.12(204~210MHz)
  - Ch.8(180~186MHz)
  - VHF 두개채널, 채널당 3개씩 총6개의 멀티플렉스 가능
- 서비스 타입
  - 오디오 서비스
    - CD Quality Stereo Audio
  - 멀티미디어 서비스
    - Video : VCD Quality(7” LCD)
    - Audio : Above FM Quality(Max. CD)
    - Data : Text and Graphic
  - Data Service
    - PAD/NPAD : News, Weather, TPEG
    - EPG, Interactive Services, Etc.
  - 속도 200km/h에서도 수신 가능할 것

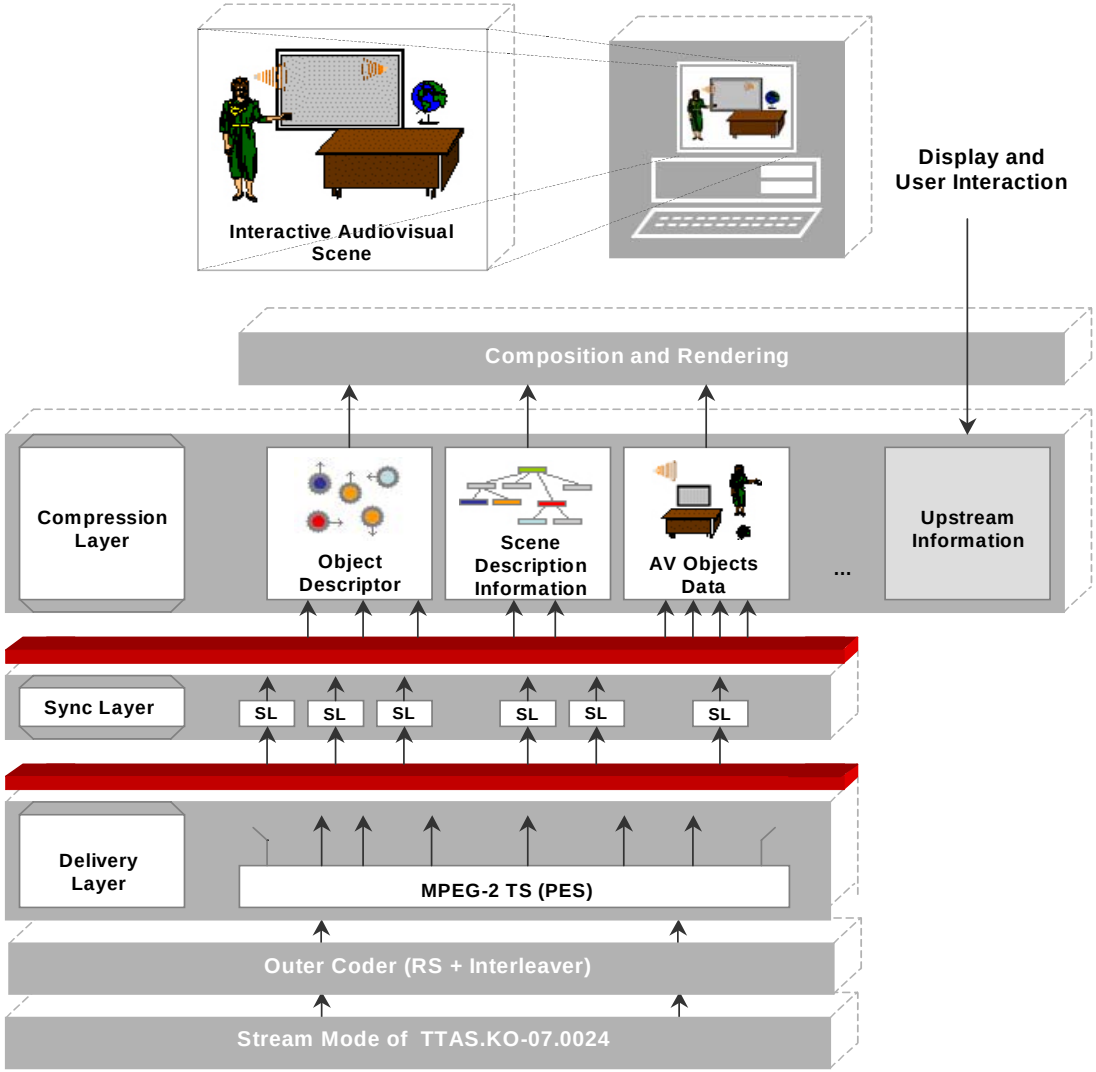
# 지상파 DMB System(1/2)

- Terrestrial DMB Standard overview
  - Base System : Eureka-147 DAB(VHF TV Broadcasting frequency)
  - Encoding : MPEG-4, 526 kbps, 7 inch
  - 3 Video and 10 Audio or other data through 1 TV channel
- Transport Standard
  - MPEG-2 system
    - PES(Packetized Elementary Stream), TS(Transport Stream)
    - PSI(Program Specific Information)
    - 14496 section
  - MPEG-4 systems
    - SL(Sync Layer)

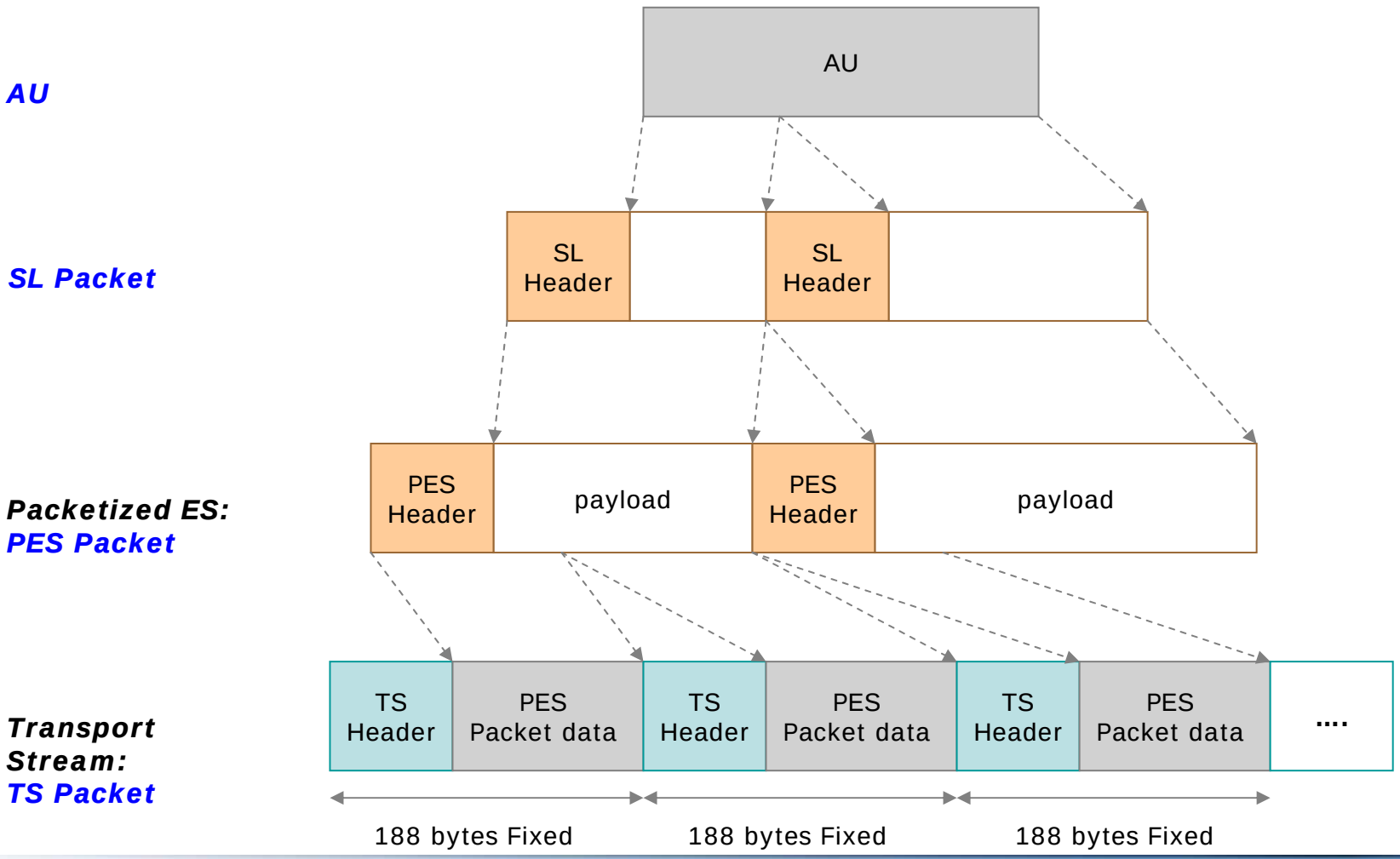
# 지상파 DMB System(2/2)

- Audio standard
  - MPEG-4 Part 3 : BSAC(Bit Sliced Arithmetic Coding)
- Video standard
  - MPEG-4 Part 10 : AVC (Advanced Video Coding)
- Data service standard
  - MPEG-4 system
    - 장면 기술 규격
      - ISO/IEC 14496-1 Core2D @Level 1
    - 그래픽 데이터 규격
      - ISO/IEC 14496-1 Core2D @Level 1

# 지상파 DMB : High level architecture



# Mapping of AUs to TS packet



# MPEG-4 System Core2D profile

- 비디오 위에 간단한 2차원 그래픽 오버레이
  - 원,사각형,문자,다각형
- 2차원 그래픽의 애니메이션 효과
- 객체들간의 상호작용
  - 웹 링크
  - E-commerce
  - BIFS-Update 기능

# DMB를 이용한 데이터 방송서비스의 예

- 정보제공 서비스(뉴스, 날씨, 교통정보, 증권 등)
  - 단말기를 통해 실시간 뉴스, 날씨정보, 교통정보, 증권 시황 등을 알아 볼 수 있음


< 데이터 방송 정보제공 서비스의 예>

# DMB를 이용한 데이터 방송서비스의 예

- TV 시청 및 제품 구매
  - 단말기를 통해 방송 시청을 하며 여배우의 액세서리 등을 단말기 키조작을 통해 구매 주문할 수 있음

# 요 약

- 객체 기반 멀티미디어 시스템의 개요와 기술 소개
- MPEG-4 시스템 툴 소개
- 지상파 DMB 기술의 소개

# References


- 2005 지상파DMB 기술 심층 워크샵
  - 지상파 DMB 대화형 기술 - MPEG-4 BIFS(netntv 임영권)
  - PC 기반 지상파 DMB 수신 S/W기술(서울시립대학교 김용한)
- 한국방송공학회 방송기술국체워크샵, 2004.5.
  - Introduction to Advanced Audio Coding(픽스트리,김연배)
- MPEG-2 표준과 디지털방송(세종대학교 최석림)
- 2005 MPEG 포럼 총회 및 워크샵
  - MPEG 표준화 절차 및 활동 가이드라인(한국전자통신연구원 김규헌)
- 방송공학회
  - DMB 다중화기술(서울시립대학교 김용한)
- 초단파 디지털 라디오 방송(지상파DMB) 비디오 송수신 정합표준 TTA
- 지상파 DMB 서비스 - 데이터 방송을 중심으로(YTN 기술연구소 준비팀, 2004)
- 지상파 DMB 미디어 재생기 (한국전자통신연구원 기명석)
- 2004 MPEG 포럼 워크샵 발표 자료
- 지상파 이동 멀티미디어 방송(한국전자통신연구원 이현)

- ISO/IEC JTC1/SC29/WG11 published documents