



어플리케이션 개발

- 가상 기계
  - 미들웨어는 반드시 JVM (Java Virtual Machine)을 가상기계로 사용
  - 응용서비스는 기본적으로 STB의 CPU나 칩셋 같은 하드웨어나 운영체제 (Operating System: OS)와 관련 없이 동일하게 수행하여야 하는 Java Byte Code로서 JVM을 통해 실행되어짐
  - STB의 CPU, OS, 그래픽스, 하드웨어 등에 적합하게 설계됨
  - 상호 Integration은 C 언어를 통하여 이루어짐

- Demux 제어부
  - 스트림으로 전송되어 온 데이터를 분류하는 기능
  - 하드웨어적으로 처리됨
  - 데이터방송에 필요한 프로그램 및 서비스에 대한 정보도 모두 Demux를 통해 처리되기 때문에 Demux를 제어하는 부분은 필수적
  - MPEG-2 섹션을 Demux 하드웨어에 요구하여 얻어내고, 얻어낸 정보를 Carousel, SI/AIT 처리부 등으로 전달하는 기능
- 튜너 제어부
  - 채널 및 서비스 선택 등을 수행하는 부분
  - 물리적 (Physical) 채널의 선택은 튜너부와 연동하여 수행

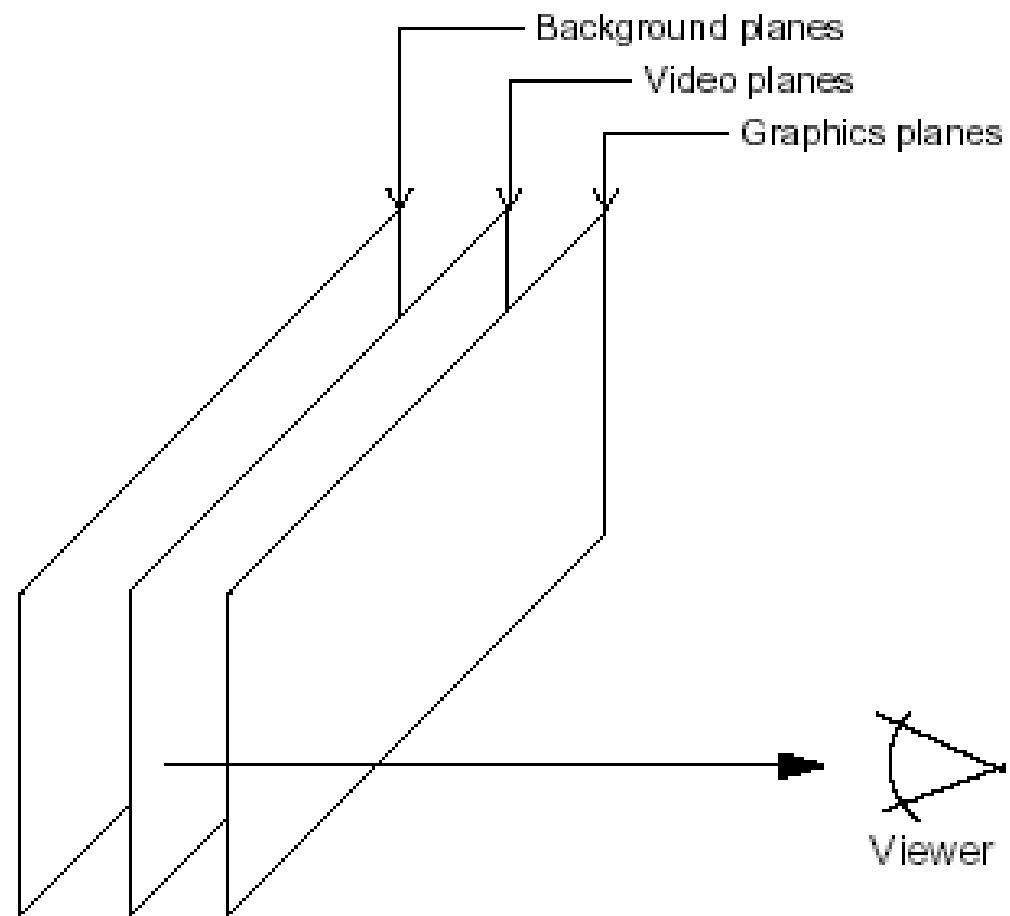
- CA(Conditional Access) 처리부
  - POD의 제한 수신 인터페이스(POD-CA Interface)를 이용하여 사용자의 채널, 서비스 등에 대한 접근 권한을 관리
  - 필요한 경우 사용자로부터 접근을 위한 PIN, 암호 등을 얻음
  - 권한에 따라 디스크램블러와 연동하여 콘텐츠를 사용자에게 보여줄 것인지를 결정
  
- Carousel 처리부
  - 응용서비스 데이터는 Carousel로 전송됨
  - 데이터 스트림 내부에 있는 Carousel 데이터를 모아 관리하면서 디렉터리와 파일이라는 추상적인 형태로 응용서비스나 다른 미들웨어 모듈들이 데이터에 접근할 수 있게 해 주는 기능

- SI/AIT 처리부
  - MPEG, DVB, MHP, OCAP 등에 의해 정의되어 있는, A/V 프로그램, 응용서비스 등 방송을 이루는 각 기본 단위들에 대한 기본적인 정보를 실어 나르는 테이블들을 처리하여 고급 정보로서 다른 모듈들이 사용할 수 있도록 제공
- 그래픽스 처리부
  - 도형이나 글자를 사용자에게 보이는 기본적인 그래픽스 기능 제공
  - 그래픽 객체로서 MHP 및 OCAP 에서 요구하는 HAVi (Home Audio Video Interoperability) level 2 User Interface의 위젯 (Widget)들을 지원
  - 각 위젯들 간의 상하 처리에 의한 가시화, 이벤트 전달 등을 처리

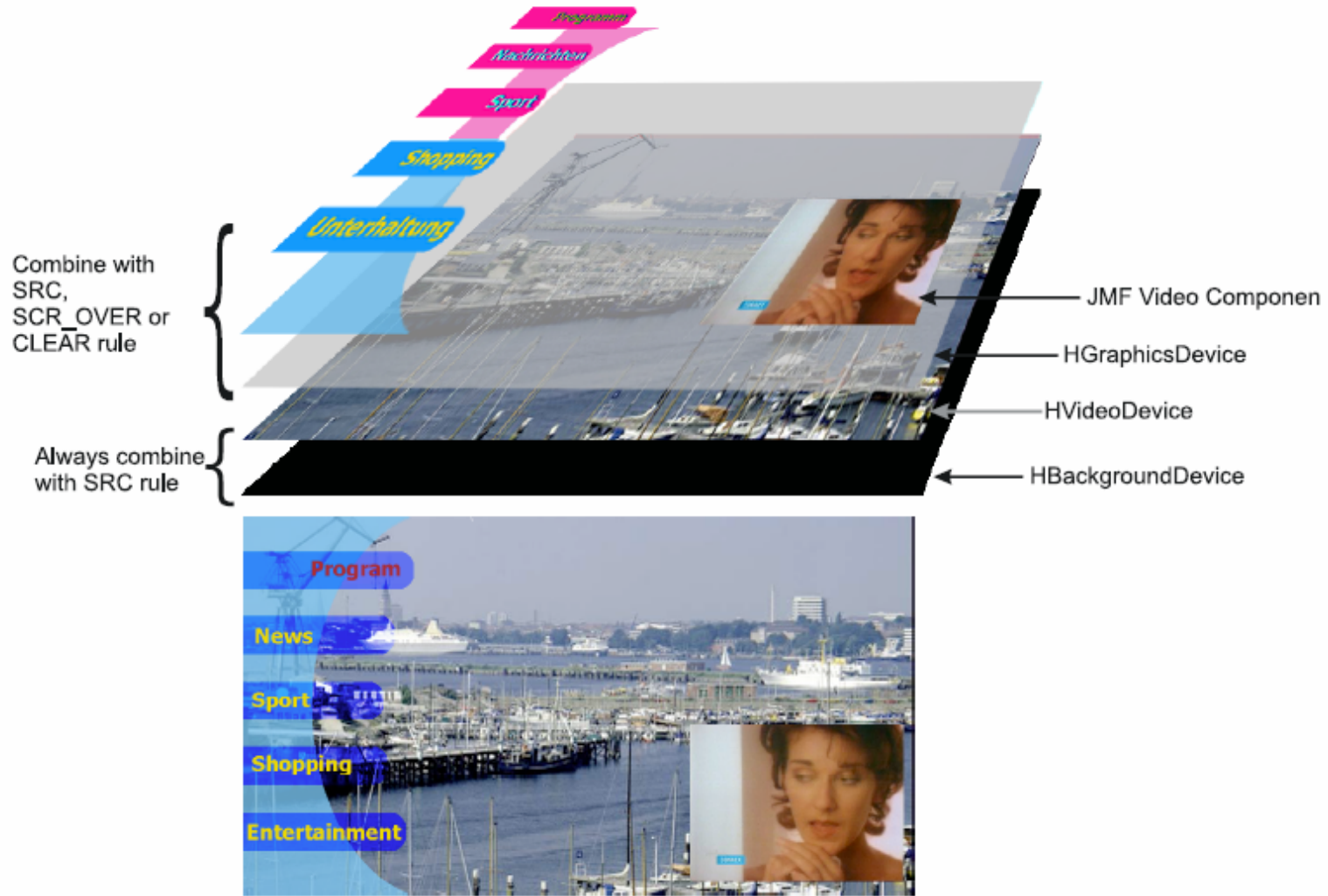
- Application 관리자
  - SI/AIT 처리부에 의해 방송 스트림으로 응용서비스가 방송됨을 전달 받으면 이에 따라 방송된 응용서비스를 가상기계상에서 동작시키거나 중지시키는 역할을 수행
- 자원 관리자
  - STB H/W 자원들이 응용서비스나 미들웨어 모듈에 의해 공유되며, 많은 경우 자원들은 그들의 희소성이나 독점적으로 사용되어야 하는 성질 때문에 관리되어야 하며 MW의 자원 관리자는 이러한 기능을 수행
- TCP/IP 처리부
  - PSTN/Ethernet/OOB RC/DOCSIS 등을 통해 외부 네트워크와의 연동을 지원
  - 응용서비스에게 TCP/IP를 이용한 안전한 (Secure) 소켓 (Socket) 연결을 지원

# OSD Layer (1)

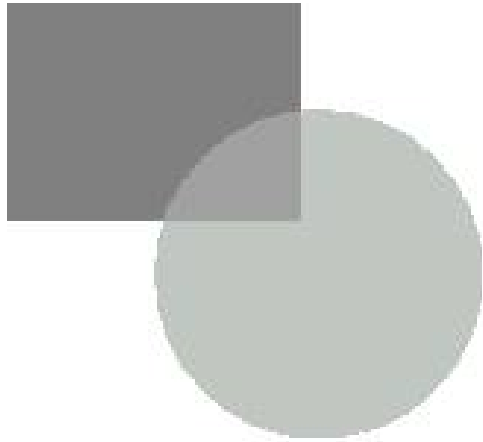
---



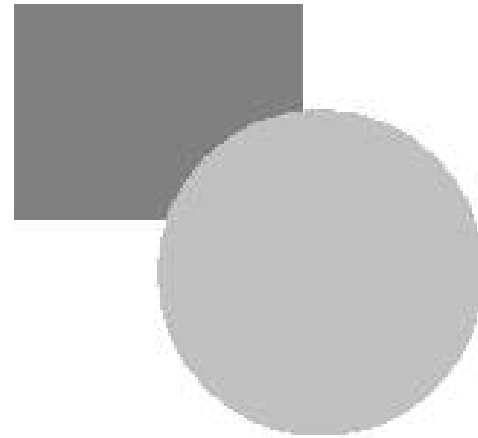
# OSD Layer (2)







SRC\_OVER



SRC

## Key Event

---

Constant name	Key	Key code (if standardized)
VK_UP	up arrow	not standardized
VK_DOWN	down arrow	not standardized
VK_LEFT	left arrow	not standardized
VK_RIGHT	right arrow	not standardized
VK_ENTER	enter (also known as select or OK)	not standardized
VK_0 to VK_9	number keys	48 - 57
VK_TELETEXT	teletext key	459
VK_COLORED_KEY_0	first coloured key	403
VK_COLORED_KEY_1	second coloured key	404
VK_COLORED_KEY_2	third coloured key	405
VK_COLORED_KEY_3	fourth coloured key	406

For more information, see the [HAVi HRcEvent](#) or [java KeyEvent](#)

- 자바 AWT 모델에서 컴포넌트는 포커스를 가질경우만 리모콘 이벤트를 받을수 있음
- MHP EPG 제약사항
  - 비디오 시청 상태에서는 화면에 아무것도 보이지 않음
  - 따라서, 사용자 입력을 받을 수 없음
    - DCA 를 통한 채널 이동이 안됨
    - 볼륨 조절을 할 수 없음
    - HOT 키를 컨트롤 할 수 없음
- org.dvb.event package 는 어플리케이션이 AWT 이벤트 매커니즘 전에 키 이벤트를 받을 수 있는 방법을 제공
- 어플리케이션은 UserEventListener 을 등록하여 키 이벤트를 받음
- 이것은 UserEvent 객체를 사용함
- UserEvent 를 사용하기 위해서, 어플리케이션은 UserEventRepository 를 정의해야 함
  - ➔ Shared User Event

## UserEventRepository 정의

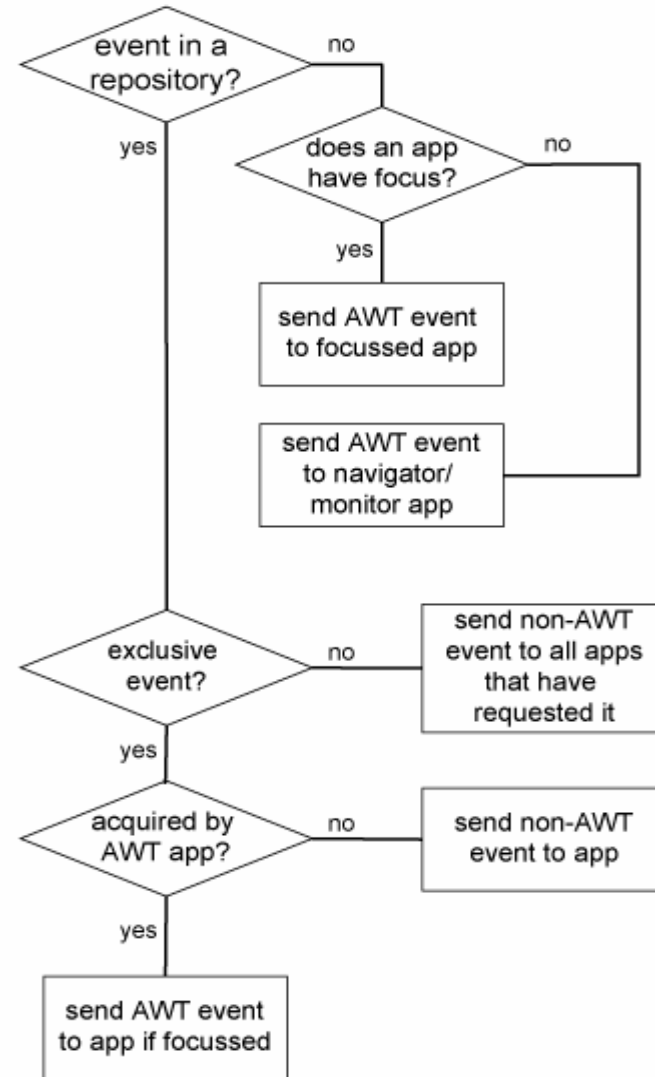
---

```
public class UserEventRepository {  
  
    public UserEventRepository (String name);  
  
    public void addUserEvent (UserEvent event);  
    public UserEvent[] getUserEvent ();  
    public void removeUserEvent (UserEvent event);  
  
    public void addKey (int keycode);  
    public void removeKey (int keycode);  
  
    public void addAllNumericKeys();  
    public void addAllColourKeys();  
    public void addAllArrowKeys();  
  
    public void removeAllNumericKeys();  
    public void removeAllColourKeys();  
    public void removeAllArrowKeys();  
  
}
```

- ❖ Before an application can receive User Events, it must define it.
- ❖ It defines the group of events that the application wishes to receive.

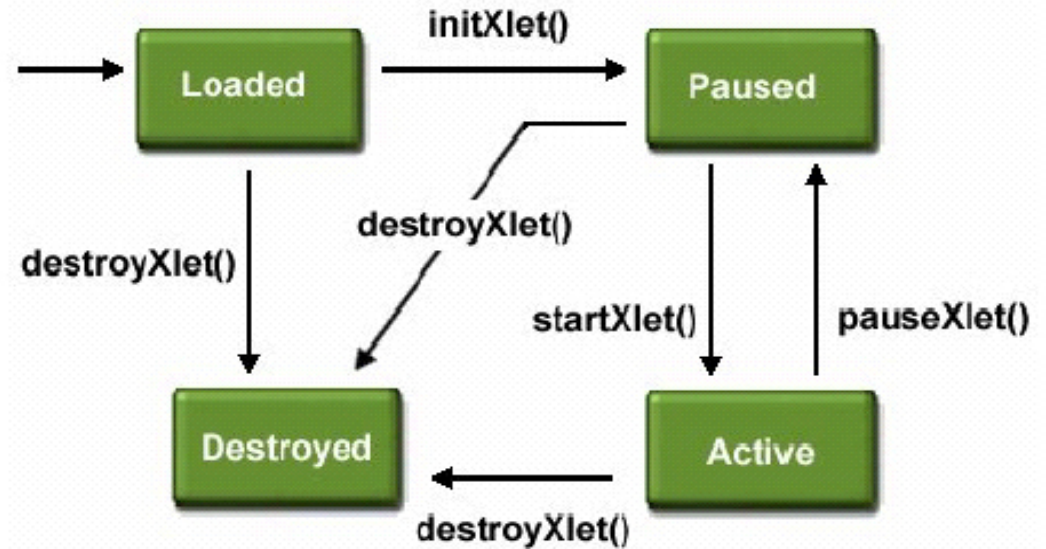
# Exclusive User Event

- Shared User Event 문제점
  - 어플리케이션이 포커스 정책에 의해 비밀번호 등을 필요로할 때
  - 다른 어플리케이션이 동일키를 Shared User Event 로 등록할 경우
  - 키 충돌 문제가 발생을 함
- EventManager 는 다음 함수를 가짐  
addExclusiveAccessToAWTEvent()

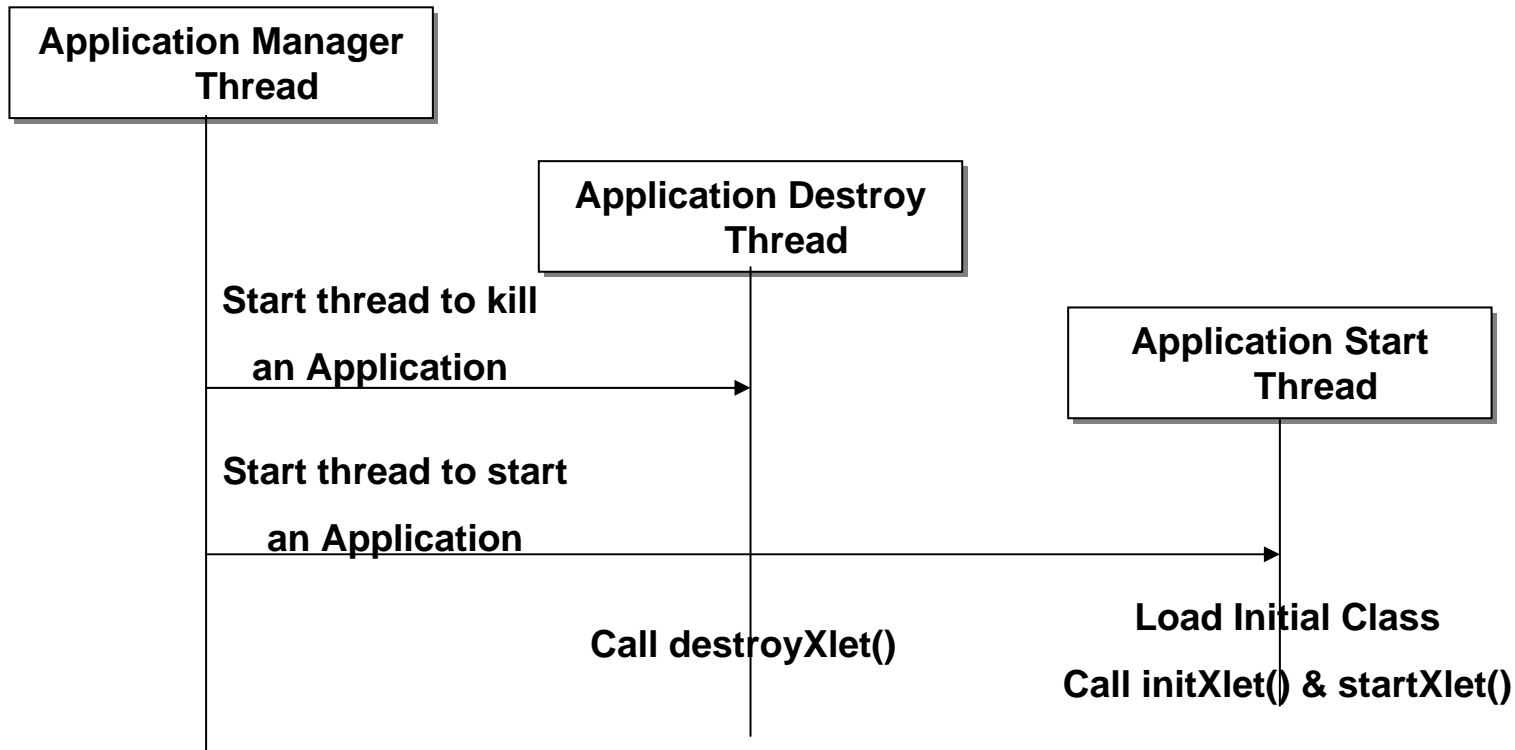


# Application Lifecycle (1)

- Loaded
  - Loaded on memory
  - Not initialized
- Paused
  - Initialized
  - Uses minimum resource
- Active
  - In normal operation
- Destroyed
  - Releases resource
  - Terminate



## Application Lifecycle (2)



- Application Manager destroy an Application and start an Application simultaneously.
- Therefore, the graphics of two Application can be overlapped.
- In `destroyXlet()`, do not anything as follows.
  - Do not update graphics, Do not draw Iframe.
  - Do not change video size, Do not control AV.

# 비디오 Control

- Use `javax.tv.media.AWTVideoSizeControl`
  - Scaling and resizing 다음의 좌표에서 이루어짐 ((0,0) ~ (720, 480))
  - `AWTVideoSize(Rectangle, Rectangle)` 를 사용
  - First: source rectangle.
  - Second: destination rectangle to be displayed.


safety zone

**PERFORMANCE BIKE** exit

Aerodynamic shape  
**SHOEI x-sp II**


units	price
0   100	\$550
100   300	\$500
300   500	\$450

**now**



TOP SPEED: 167.4 MPH

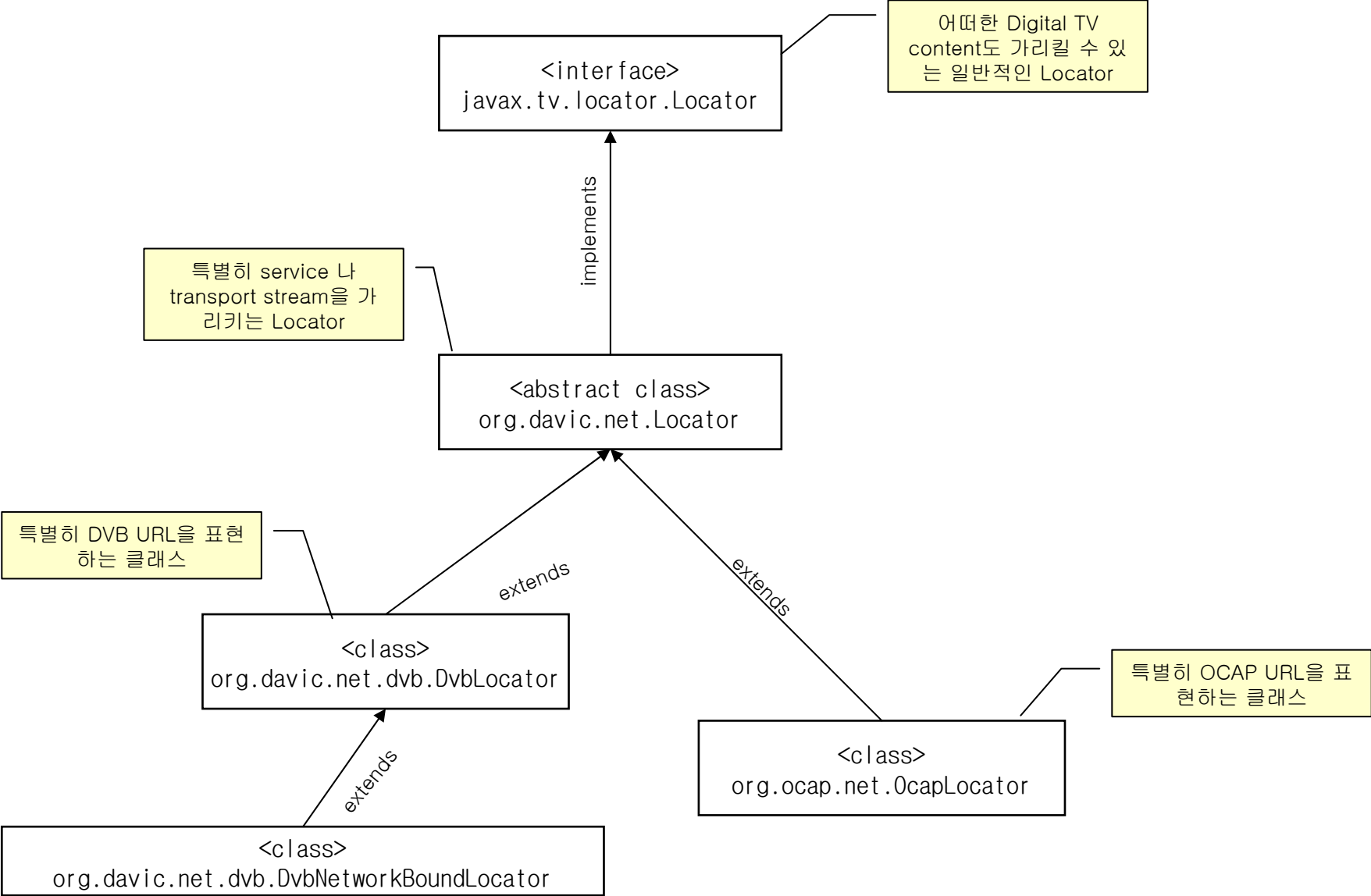
Bigger the group, lower is the price

	List Price	\$ 550.55	<b>Join This !</b>
	Current Price	\$ 495.00	
	Time Remaining	0:59:05	

8 FINEST BIKE GRAND PRIX 2001 GROUP BUYING



# Locator (1)



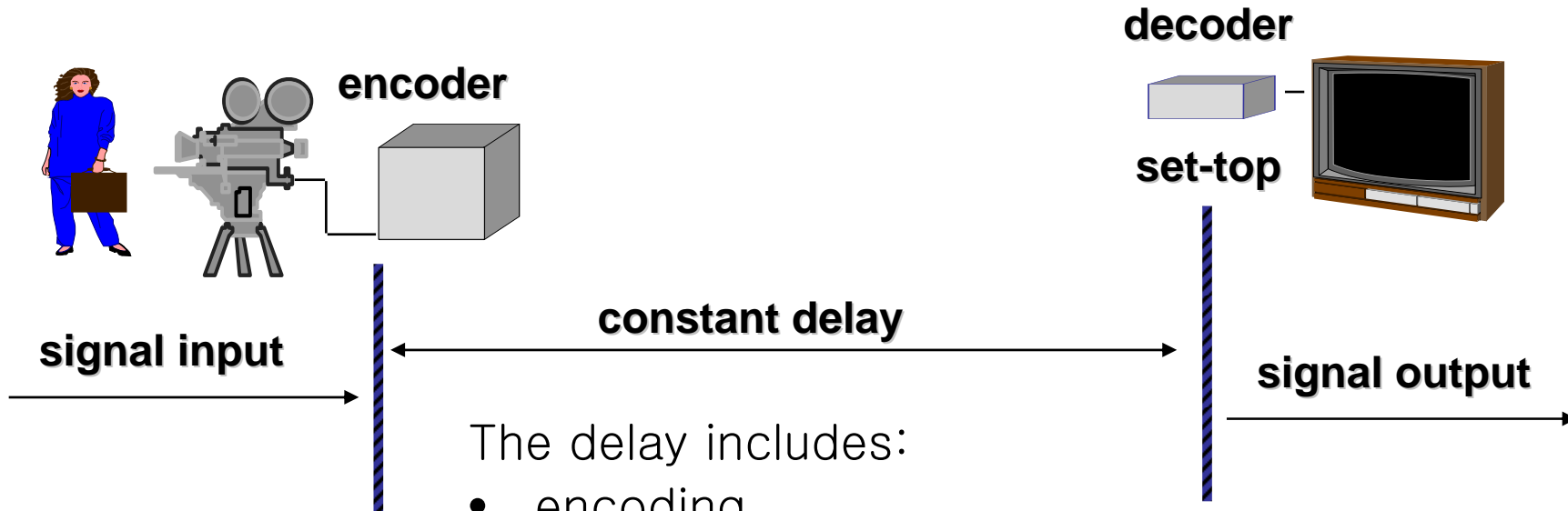
# 오디오 Control

- Use  
javafx.tv.media.MediaSelectControl



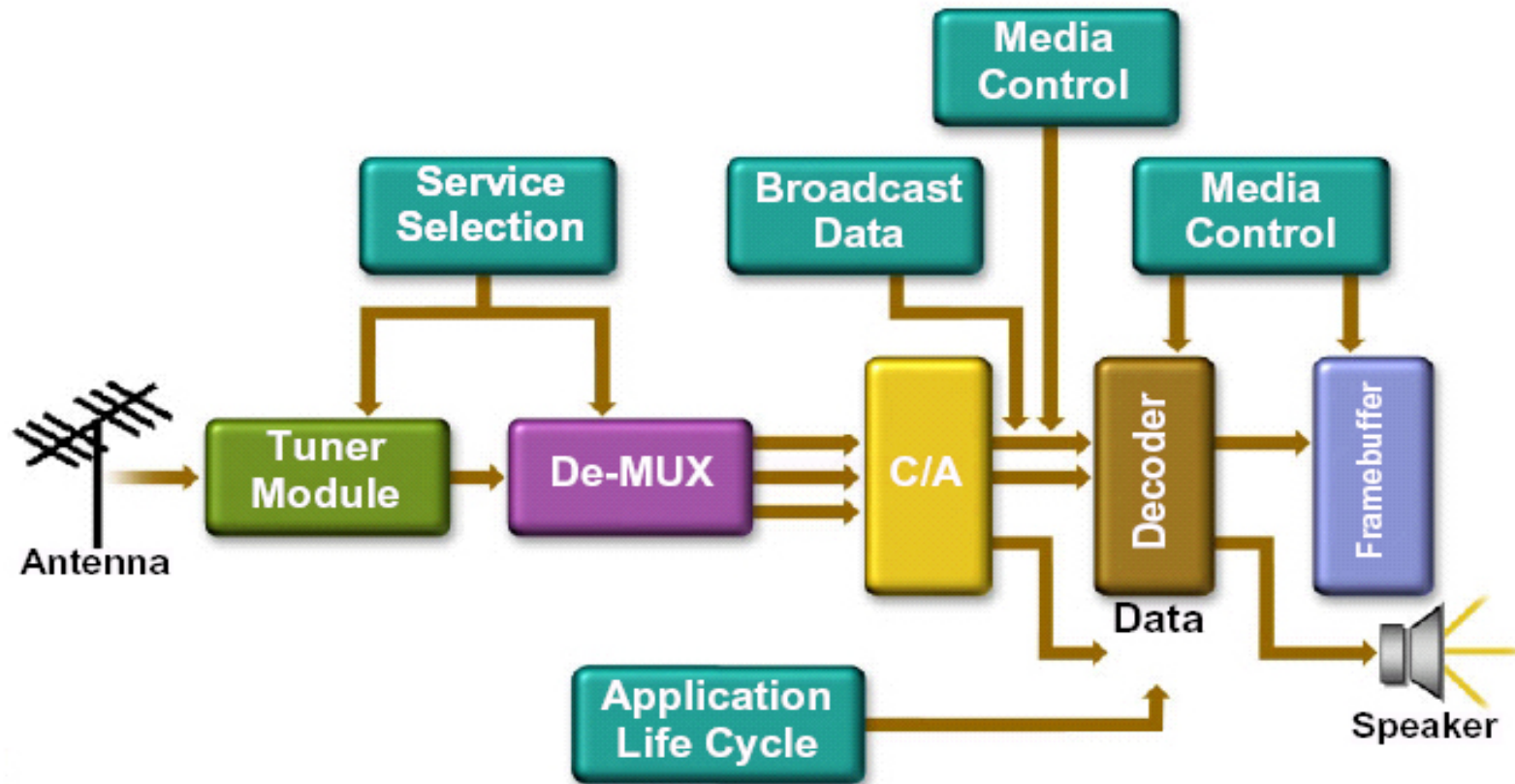
- 업데이트가 자주 발생하는 경우 또는 자주 발생하지 않는 경우
  - DSMCCObject마다 리스너를 걸어 업데이트 여부를 데이터 파싱할때마다 확인을 함
  - 즉, 업데이트가 되어있으면 파싱을 하고 업데이트가 안되어 있으면 기존에 파싱했던 정보를 그대로 이용함
  - 매번 start시 마다 해당 씬에서 필요한 데이터 파싱을 시도함. 업데이트 안된 정보는 파싱하지 않음

- OOB Section이용할 경우
  - Out Of Band의 영역에 데이터를 필터링하는 것으로 Section Filtering을 함
  - 데이터가 안정적 (서버와 채널에 의존적이지 않다)
  - 데이터 송신주기를 관장하는 부분은 송출단에 있으므로 데이터 수신 주기를 신경쓸 필요가 없음
  - 데이터크기에 제한이 있음: 1.5kbyte
- OC 파일을 이용할 경우
  - 특정채널에 할당된 OC데이터에 대해 attach를 하여 데이터를 받음
  - 만약 1개 이상의 채널에서 다른 OC데이터를 받을 때는 두개 이상의 채널간에 ChannelEvent가 올때마다 채널 번호를 비교하여 로드함
  - 다른채널로 들어가면 detach를함



The delay includes:

- encoding
- encoder buffering
- multiplexing
- communications or storage
- demultiplexing
- decoder buffering
- decoder
- presentation



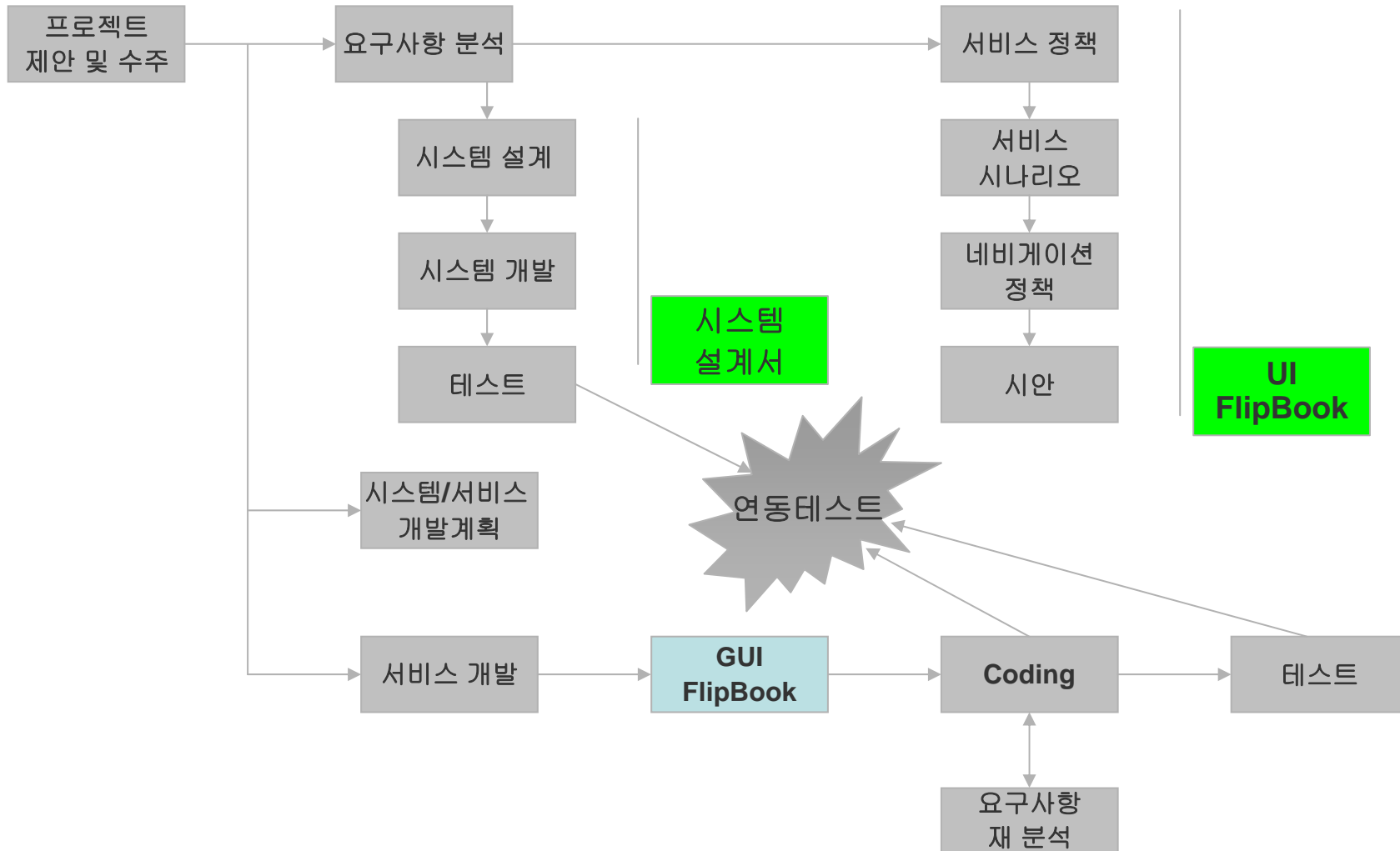
- Application Life Cycle
  - Xlet 컨트롤
- Service Information (SI)
  - 채널 및 프로그램 정보 제공
- Service Selection
  - 채널 튜닝
- Broadcast Data
  - 파일 데이터 접근
- Media Control
  - 오디오 & 비디오 컨트롤

- HAnimation
  - The HAnimation class is a user interface component used to display a sequence of images (as HStaticAnimation) which additionally enables a user to navigate (focus) upon it.
- HIcon
  - The HIcon class creates an a graphical image.
- HText
  - The HText class is used to display static, read-only, navigable text.
- HListGroup
  - The HListGroup manages a set of HListElement and presents these elements to represent a scrollable list.
- HMultilineEntry
  - The HMultilineEntry class is used to receive multiple lines of alphanumeric entry from the user.



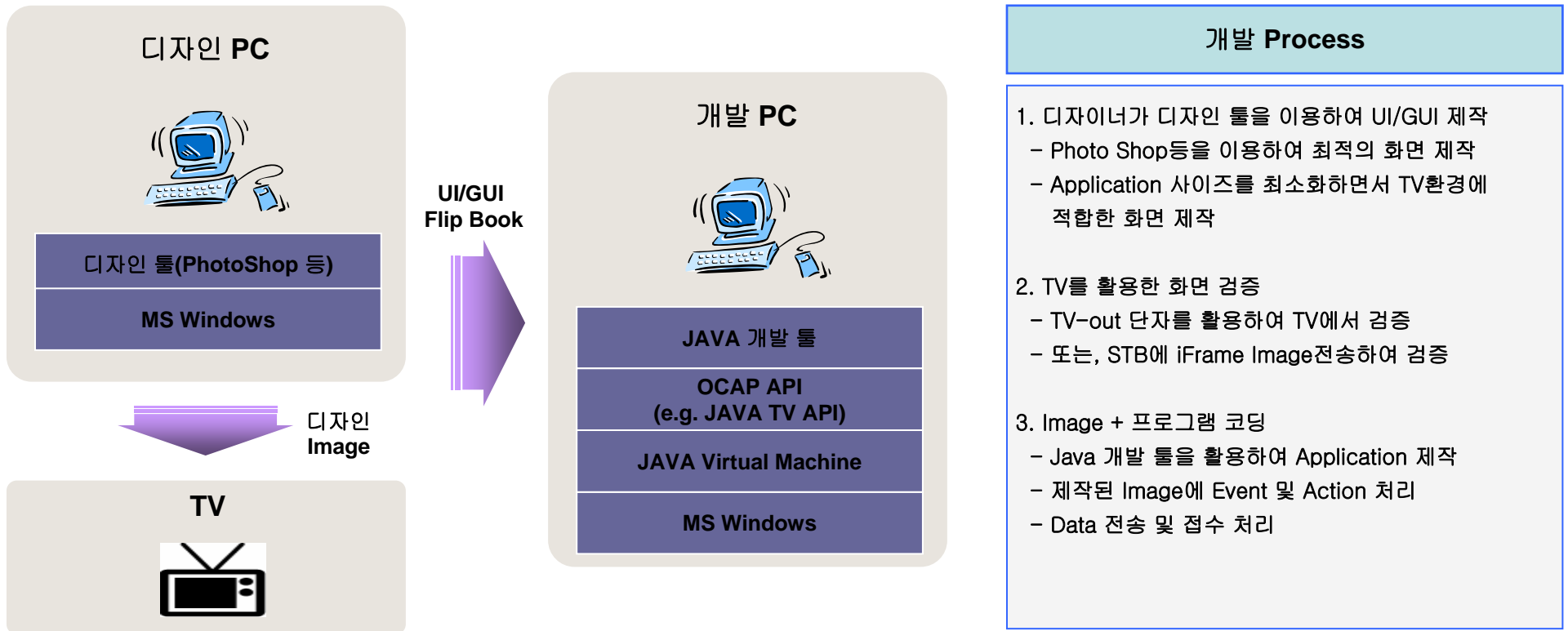
- HSinglelineEntry
  - The HSinglelineEntry class is used to receive a single line of alphanumeric entry from the user and can also be used for password input.
- HRange
  - The HRange component is used for displaying a value which is within a fixed range.
- HGraphicButton/HToggleButton/HTextButton
  - The HGraphicButton class is used to represent a conventional push-release button used for user actions.
  - The HToggleButton class creates a “check box”, or with the support of the HToggleGroup class, “radio buttons”.
  - The HTextButton class is a conventional push-release textual button to be used for user actions.

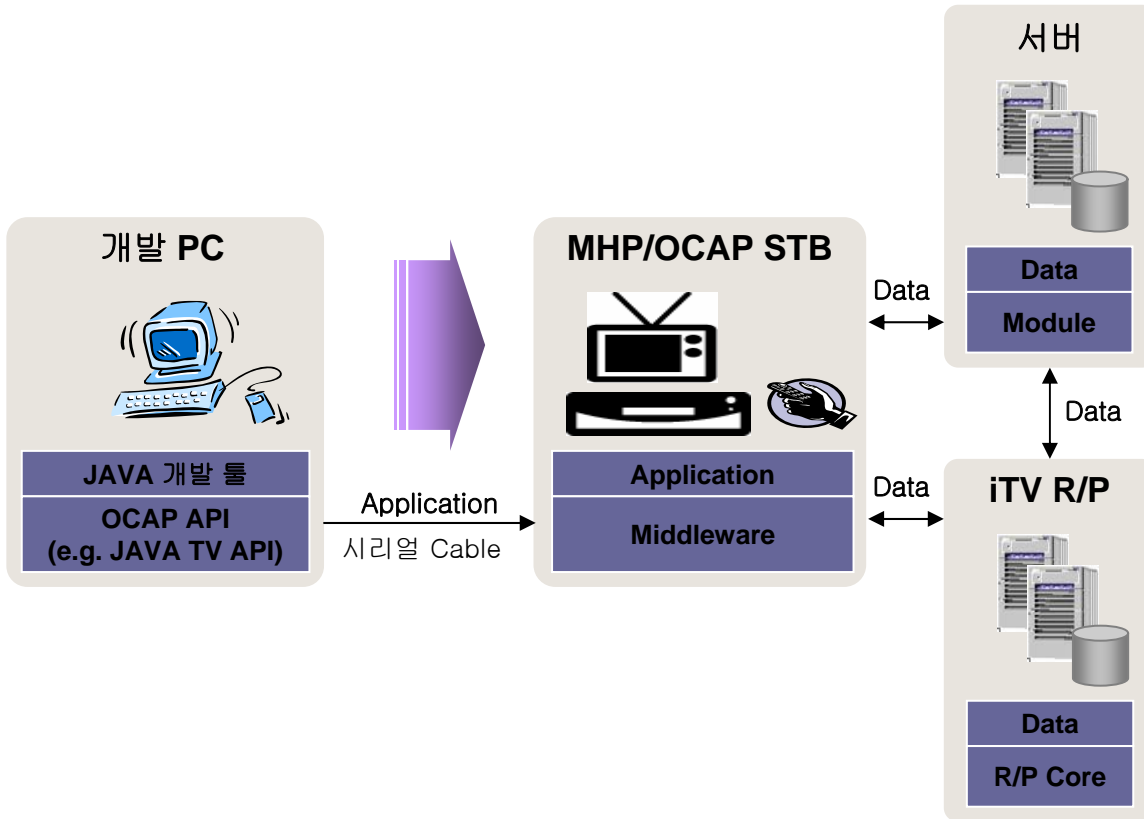
# 개발 프로세스



# 어플리케이션 형태 결정

	Bound App.	Unbound App.
개요	<ul style="list-style-type: none"> <li>채널을 통하여 다운로드되는 Application</li> </ul>	<ul style="list-style-type: none"> <li>채널과 상관없이, Flash Memory 상에 탑재되는 Application</li> </ul>
장점	<ul style="list-style-type: none"> <li>높은 서비스 안정성</li> <li>변동 사항이 생겼을 경우 업데이트 용이</li> <li>업데이트 때마다 통합테스트 불필요</li> <li>Unbound 에 비해서 App. Size 증설 용이</li> </ul>	<ul style="list-style-type: none"> <li>어떠한 채널상에서도 사용 가능</li> <li>빠른 로딩 속도 (1초 이내)</li> </ul>
단점	<ul style="list-style-type: none"> <li>타 채널과 연계 난이</li> <li>느린 로딩 속도 (채널변경시간 + 다운로드시간 + 구동시간 -&gt; 3초 이상)</li> <li>채널 변경 후 재 접속 시, 신규 다운로드로 인한 이전 입력 중인 Data 저장 불가</li> </ul>	<ul style="list-style-type: none"> <li>낮은 서비스 안정성</li> <li>업데이트 때마다 통합 테스트 필요 (서비스 변경이 용이하지 않음)</li> <li>플랫폼 사업자에 따라 App. Size 제약 (Flash Memory 할당이 불가할 수 있음)</li> <li>플랫폼 사업자의 Monitor App. 수정 개발 필요</li> <li>비용 상승</li> </ul>

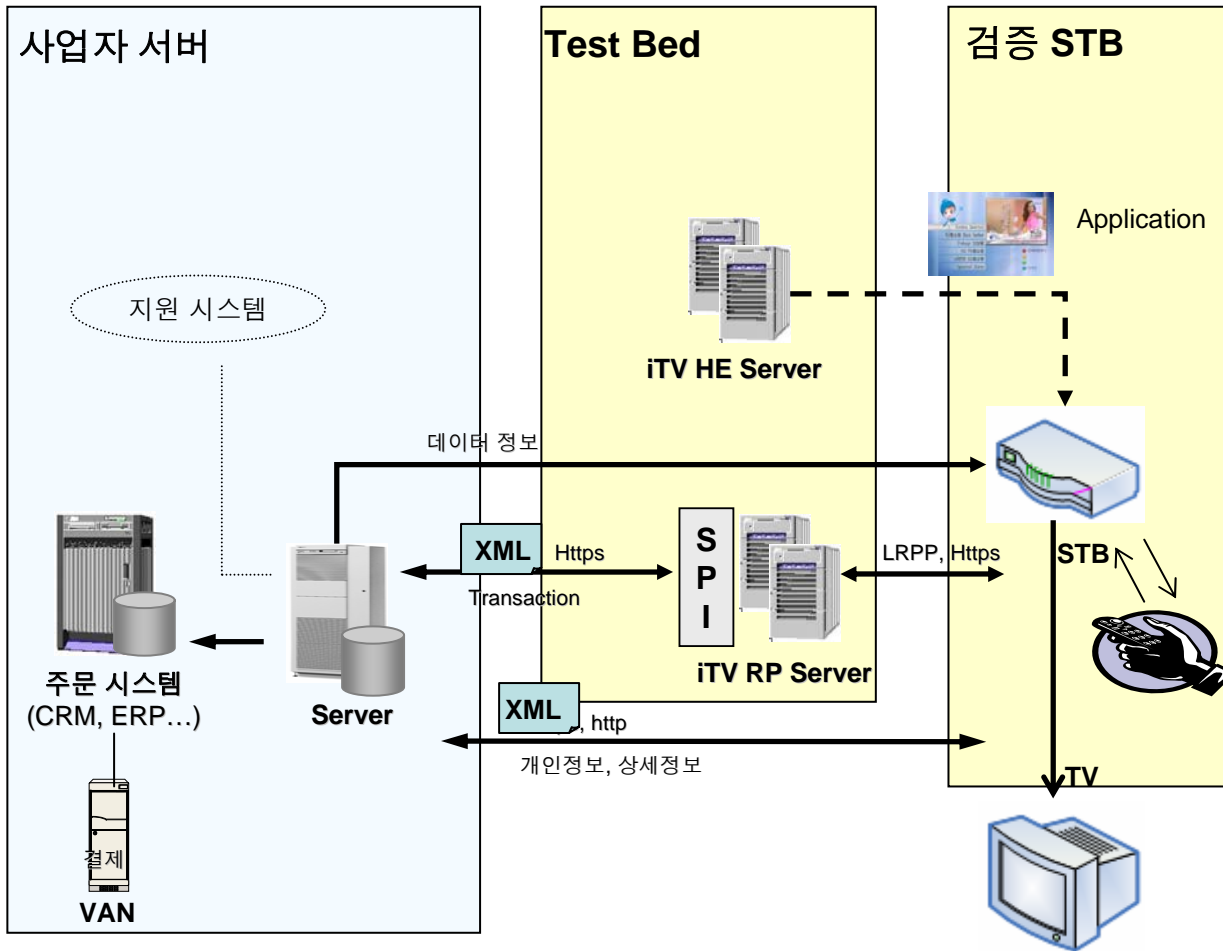




## Test Bed를 통한 검증 Process

1. STB에 Application 이식
  - 개발 PC에서 시리얼 Cable등을 통하여 STB에 Application을 download
  - STB 상에서 Application이 정상적으로 로딩되는지 체크
2. 서버와의 통신을 통한 Transaction 검증
  - Return 채널을 통한 정보를 조회
3. R/P 서버
  - R/P 서버를 통하여 제공 받는 정보(가입자 정보) 수신확인

# 플랫폼 테스트 베드



## Test Bed를 통한 검증 Process

1. STB에 Application 이식
  - iTV H/E에서 Application을 할당된 채널로 송출
  - STB 상에서 Application이 정상적으로 로딩되는지 체크
2. 서버와의 통신을 통한 데이터 검증
  - Return 채널을 통한 데이터 상세 정보를 조회
3. R/P 서버
  - R/P 서버를 통하여 제공 받는 정보(가입자 정보) 수신확인

# 테스트 프로세스

