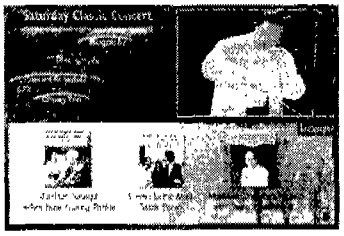
Introduction to ATSC DASE

(DTV Application Software Environment)

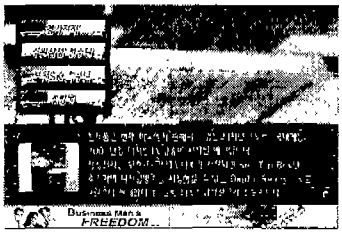
이 광 기
kklee@samsung.com

Program에 연동된 데이터 서비스

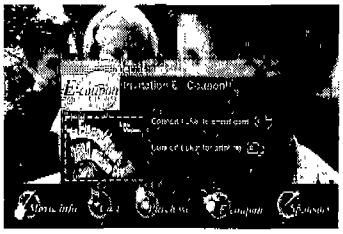


• Declarative application

• Java (Xlet) application

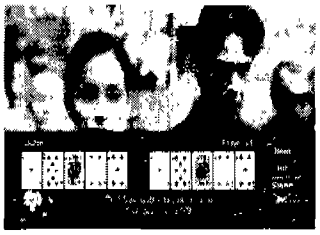


• Animation 을 이용한 광고

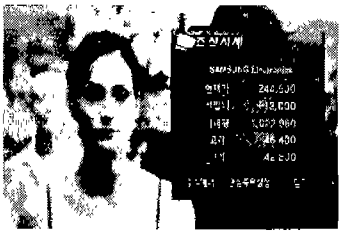


• E-Coupon Service

Program과 독립적인 데이터 서비스



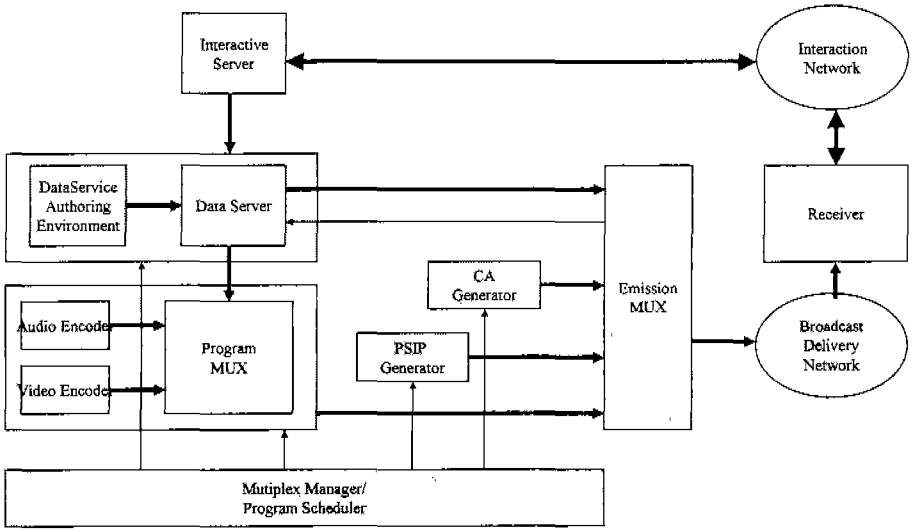
• 게임: Java application



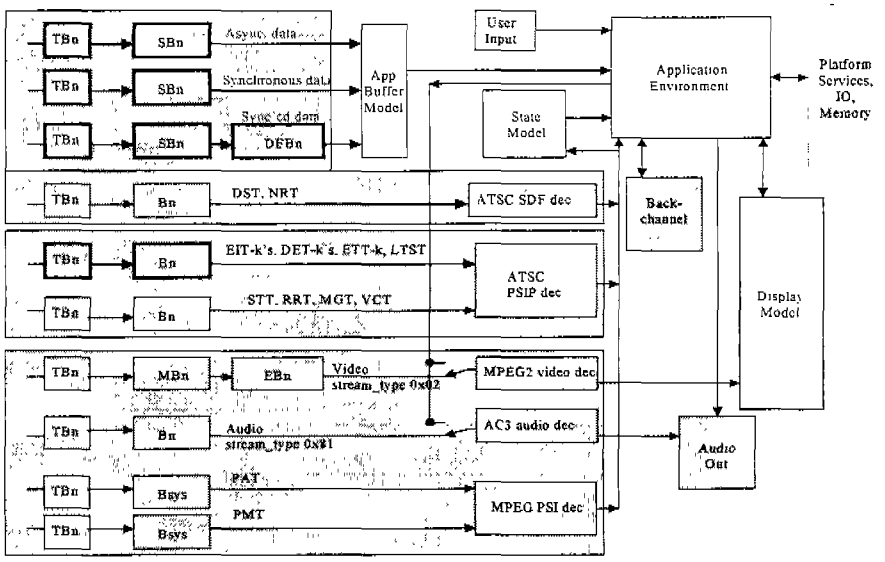
• 주식: Java application

- **MP3 Music Download** 서비스
- **Smart Card**를 이용한 결제
- **Smart Media**를 이용한 타제품 활용
- **Network** 매체로서의 **DTV** 방송의 예

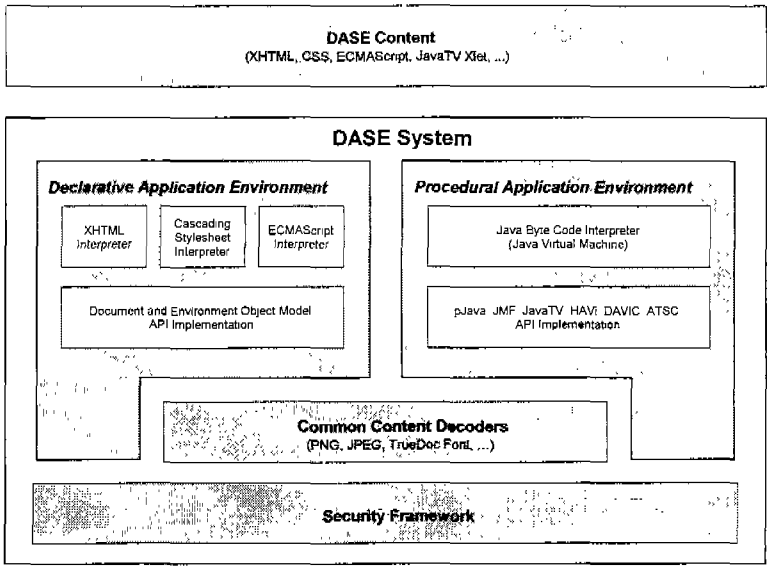
Interactive Data Broadcast Env.



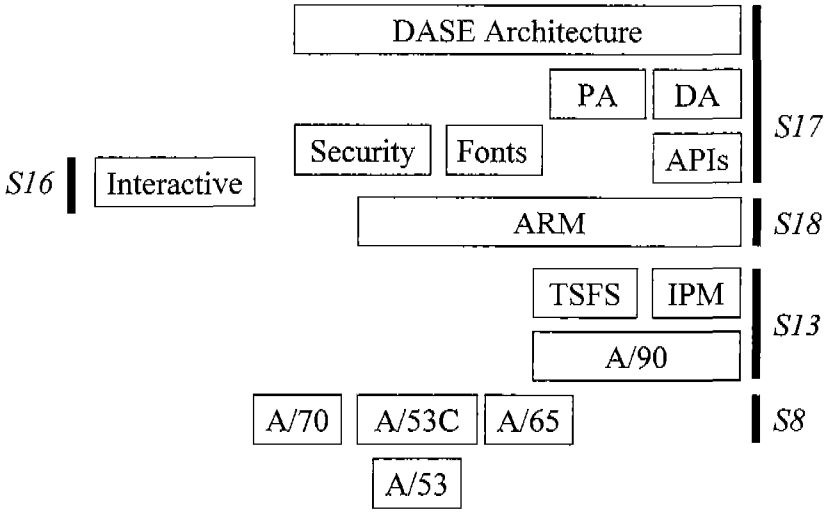
Receiver Reference Model Data Flow



DASE Architecture



ATSC Technical Documentation Efforts



DASE (ATSC T3/S17) Standard

- Ballot passed by technology group (T3)
  - Part 1: Introduction, Architecture, and Common Facilities
  - Part 2: Declarative Applications and Environment
  - Part 3: Procedural Applications and Environment
  - Part 4: Application Programming Interface
  - Part 5: Portable Font Resource
- Under development
  - Part 6: Security (Under development)
  - Part 7: Application Delivery (Under development and moved out)
  - Part 8: Conformance (Under development)
  - Part 9: Compliance Testing (Under development)

Facilities common to both DA and PA

- Graphics Content : image/jpeg (baseline mode)
  : image/png
- Non-Streaming Video : video/mng (MNG-LC)
- Non-Streaming Audio : audio/basic (PCM)
- Streaming Video Content : video/mpeg (must adhere to ATSC A/53)
  : video/mpv (must adhere to ATSC A/53)
- Streaming Audio Content : audio/ac3 (must adhere to ATSC A/53)
- Font Content : application/font-tdpfr (TrueDoc PFR)
- Archive Content : application/jar
- Trigger Content : application/dase-trigger

❖ **Non-streaming video/mpeg support is being discussed.**

DASE Declarative Applications

- Declarative Content Types
  - Markup Content : application/xdml+xml
  - Stylesheet Content : text/css
  - Script Content : text/ecmascript
- DASE DA – Pure
  - markup, stylesheet, script content
  - common content types
  - security content types
- DASE DA – Hybrid
  - Declarative Using Procedural Content
  - Embedded Active Object Content (Xlets) supported in DASE-1

Declarative Applications

- XDML (Extensible DTV Markup Language)
  - DTD Driver Select modules which provide orthogonal core functionality.
  - "-//ATSC//DTD XHTML Host Language XDML 1.0//EN"
  - "-//ATSC//DTD XDML FrameSet (XHTML Integration Set) 1.0//EN"
- Cascading Stylesheet (CSS) Level 2
- ECMAScript (3rd Edition)
- Document Object Model, Level 2
  - Both ECMAScript and Java Bindings

❖ **CSS-2 and DOM-2 are both subsetted and extended**

Example: DA


```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE html PUBLIC "-//ATSC//DTD XHTML Host Language XDML
  1.0//EN" "xdml-1.dtd" >
<html>
<head>
<title>Bach's home page</title>
<style> body {background-color:silver} </style>
</head>
<body>
<h1>Bach's home page</h1>
<hr />
<p>Johann Sebastian Bach was a prolific composer.</p>
</body>
```

HTML, XML, XHTML

- **XML vs. HTML**
  - **XML was designed to describe data, and to focus on what data is.**
  - **HTML was designed to display data, and to focus on how data looks.**
- **XHTML?**
  - XHTML stands for EXtensible HyperText Markup Language
  - XHTML is almost **identical** to HTML 4.01 and aimed to **replace** HTML
  - XHTML is a **stricter and cleaner** version of HTML
  - XHTML is HTML defined as an **XML application**
- **The Most Important Differences:**
  - XHTML elements must be **properly nested** and **well-formed**
  - Tag names must be in **lowercase**
  - All XHTML elements must be **closed**
  - The XHTML DTD defines **mandatory** elements

Procedural Application Environment

- Java Virtual Machine
  - JVM, 1st Edition plus Errata
  - Inner Classes Specification
  - Java Class File Versions
  - Byte Code Verification
  - Superclass/interface must be resolved when class/interface is resolved.
  - Class finalization not required.
- Procedural Content Types
  - Active Object Content : application/java
    : application/javatv-xlet
  - Application Defined Content : application/octet-stream

DASE Java APIs

- Personal Java (PJAE1.2A)
- Java Media Framework (JMF1.0)
- Java Television (Java TV 1.0)
- HAVi Level 2 UI (HAVi L2 UI 1.0.1)
- DAVIC 1.4.1 Media/Resource Services
- W3C Document Object Model (DOM)
- DASE Specific

  - **Caution: Some packages, classes and methods are modified/excluded/constrained.**

Example: Hybrid Application


```
//DAE may find xlets are embedded in DA and request AM to launch Xlet.
//Then, AM may instantiate DocumentFactory and request DAE to provide DOM
public class ImpDocumentFactory implements DocumentFactory {
     Document doc = null;      //DA who embeds xlet
     ImpDocumentFactory(url lid4xdml) { … }
     performAction(DocumentAction act) { act.run(doc); }
}
//And then, AM may launch EmbeddedXlet
public class EmbeddedXlet implements Xlet
{
     Document doc;
     initXlet(XletContext xc) {
          documentFactory = xc.getXletProperty("DOM");
          documentFactory.performAction(new MyAction ());
     }
     startXlet() {…}   // xlet accesses, mutates DOM
     public class MyAction implements DocumentAction {
          run(Document document) { doc = document; }
     }
 }
//Finally, DAE requests AM to stop and destroy EmbeddedXlet
```

Referencing Resources

- Local Identifier "lid:" Scheme
  - ***lid://name/path?query#fragment***
  - Unique identifier associated with resource, assigned by local naming authority (i.e., content author or emissions station) to refer to such broadcast resources as:
    - Application
    - Application Tap
    - Module
    - Stream
  - Headline News !
- Television "tv:" Scheme
  - Refers to "this" video program
  - body { background: url("tv:") }

Data Broadcast protocols in DASE-1

- javax.tv.carousel
  - public class **CarouselFile** extends java.io.File

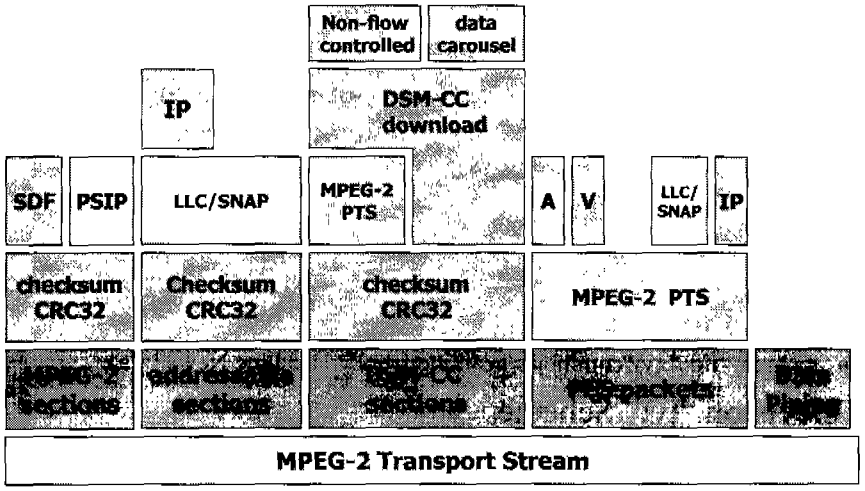
```
CarouselFile cf = new CarouselFile("lid://xyz.com/MyApp/a.jpg");
InputStream is = new FileInputStream(cf);
byte[] buf = new byte[is.available()];
is.read(buf);
```

- javax.tv.media.protocol
  - public interface **PushSourceStream2** extends PushSourceStream

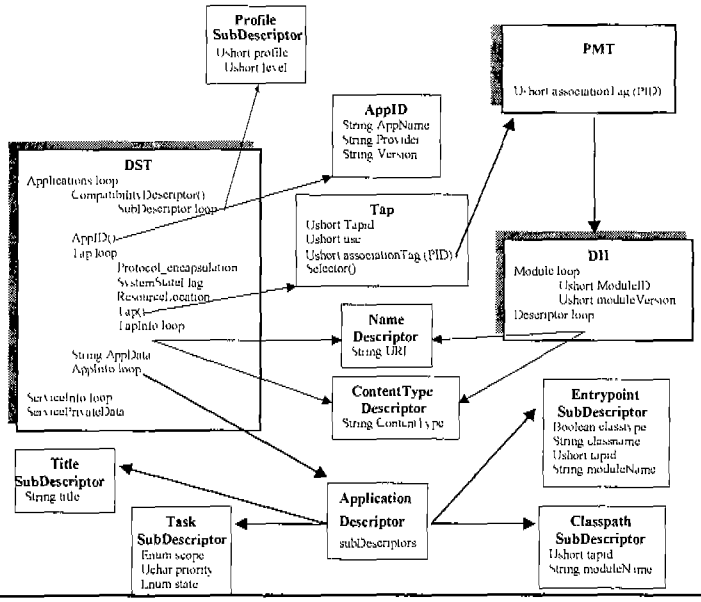
```
if((dataSource = (PushDataSource)(Manager.createDataSource
            ("lid://xyz.com/MyApp/b.str"))) == null)
 ……..
stream = (PushSourceStream2)(dataSource.getStreams())[0];
stream.setTransferHandler(this);
```

  - Built-In Data Source for "lid://xyz.com/MyApp/b.str" might be:
    - Asynchronous Data Piping : atsc.async.pipe
    - Asynchronous Data Piping, Raw Packet : atsc.async.pipe.raw
    - Asynchronous Download, No Flow Control : atsc.async.download
- javax.tv.net

Data Service Protocols



SDF Linkage (revisions in progress)



application/dase content type

- DASE Application Meta-Information Resource
  - Newly accepted content type for a root DASE Application resource.
  - Every DASE Application would contain at least two resources:
    - **root resource:** one resource of content type ***application/dase***, the
    - **initial resource:** ***application/xdml+xml*** (a DA) or ***application/javatv-xlet*** (a PA)
  - Syntax of ***application/dase*** is based on XML.
- Status
  - ***application/dase*** will require substantive changes to DASE Part 1, 2, and 3.
  - However, this change will resolve certain outstanding problems in DASE Part 6 (Security Framework) and DASE Part 7 (ARM Bindings).

Example: application/dase resource for a PA

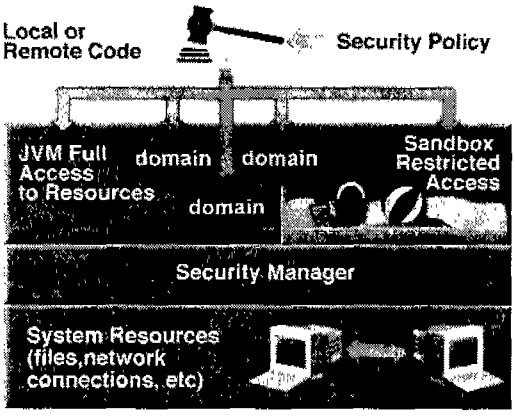

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DASE PUBLIC "-//ATSC//DTD DASE Application 1.0//EN">
<Application>
  <!-- unique application identifier -->
  <Identifier       URI="lid://xyz.com/appTwo/"/>
  <!-- application description, in English -->
  <Description xml:lang="EN">My PA, which does ...</Description>
  <!-- class path in which to search for Java classes -->
  <ClassPath        URIS="lid://xyz.com/appTwo/classes.jar"/>
  <!-- initial entity resource identifier -->
  <InitialEntity    Class="com.xyz.appTwo.Main"/>
  <!-- signature entity resource identifier -->
  <SignatureEntity  URI="lid://xyz.com/appTwo/signature.xml"/>
  <!-- require system profile/level capability -->
  <Requires         Capability="system">
    <Param          Name="PROFILE" Value="BASE"/>
    <Param          Name="LEVEL" Value="1"/>
  </Requires>
```

To be continued on the next page ...

Example: application/dase resource for a PA


```
  <Requires          Capability="graphics">
       <Param          Name="HRES" Value="640"/>
       <Param          Name="VRES" Value="480"/>
  </Requires>
  <!-- request cache capability -->
  <Requests          Capability="cache">
    <Param           Name="MinSize" Value="2M"/>
     </Requests>
  <!-- request memory capability -->
  <Requests          Capability="memory">
    <Param           Name="InitialWorkingSet" Value="1M"/>
  </Requests>
  <!-- resource cache directives -->
    <Cache  URI="lid://xyz.com/appTwo/-" Directives="no-cache"/>
    <Cache  URI="lid://xyz.com/appTwo/classes.jar" Directives="preload"/>
    <Cache  URI="lid://xyz.com/appTwo/splash.jpg" Directives="preload"/>
  <!-- application parameters (arguments) -->
  <Param             Name="ARG.0" Value="argZero"/>
  <Param             Name="ARG.1" Value="argOne"/>
</Application>
```

JDK 1.2 Security Model



Sample Code

- Permission Request

```
grant codeBase "lid://xyz.com/MyApp/"{
  permission javax.tv.service.selection.SelectPermission "*","own";
};
```

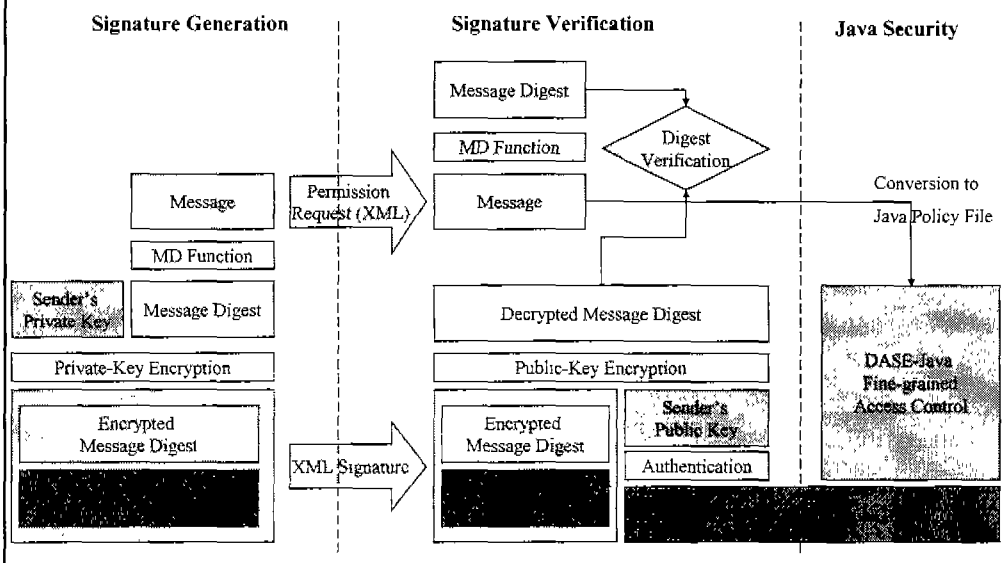
- In an Xlet from "lid://xyz.com/MyApp/"

```
try {
  serviceContext.select(locator);
} catch (SecurityException e) {
  // Ummm … my requested permission is not granted.
}
```

- ServiceContext Implementation

```
public void select(Locator locator)throws
 InvalidLocatorException, SecurityException, IllegalStateException {
  SecurityManager sm = System.getSecurityManager();
  if (sm != null) {
    sm.checkPermission(new SelectPermission(locator, "own"));
  }
  //The request for permission is granted.
  ……
```

How digital signature work?



DASE Future Work (DASE-2)

- Closed Captioning Services
- Return Channel (http, https, ...)
- Service Selection Enhancements (CRID, UPID, ...)
- Persistent Plug-In (for arbitrary multimedia contents)
- Platform Update/Upgrade Service (OS/Middleware/NativeApp)
- Concurrent Applications
- Inter-App/Xlet Communication
- Digital Signature (Certificate Services, CRL Services)
- Application Security Services (Signing, Key Management)
- Device Services (PVR, Home Networking, ...)
- Intellectual Property Management
- Declarative Enhancements (Full CSS2 Support, SMIL 2.0, XSL, ...)
- ECMAScript to Java Bridge
- Declarative Content Synthesis (by procedural content)