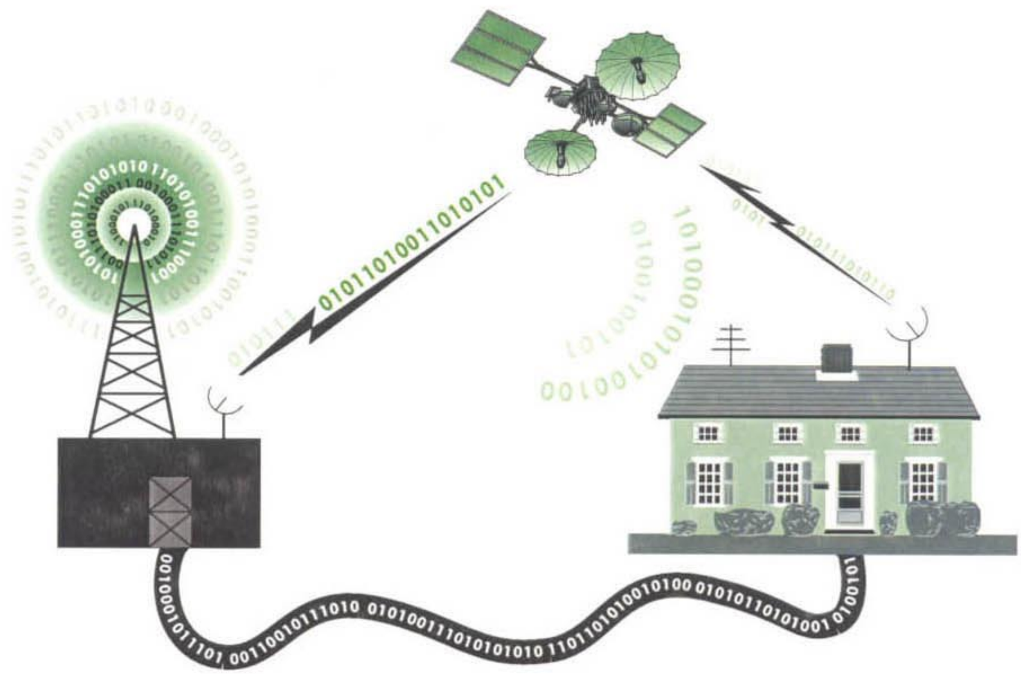
DASE Symposium 2001

NIST IR 6740

Proceedings of the

The 2nd Annual Digital TV Application Software Environment (DASE) Symposium 2001: End-to-End Data Services, Interoperability & Applications



June 19-20, 2001

National Institute of Standards and Technology
Gaithersburg, Md.

www.dase2001.nist.gov

Sponsored by:

NIST
National Institute of Standards and Technology
Technology Administration,
U.S. Department of Commerce

atsc
Advanced Television Systems Committee

NISTIR 6740

Proceedings of the

2nd Annual Digital TV Applications Software Environment (DASE) Symposium 2001: End-to-End Data Services, Interoperability & Applications

Edited by:

Alan Mink
Robert Snelick

Information Technology Laboratory

June 2001

National Institute of Standards and Technology
Technology Administration, U.S. Deportment of Commerce

U.S. Department of Commerce
Donald L Evans, Secretary

National Institute of Standards and Technology
Karen H. Brown, Acting Director

Table of Contents

Session 3: DASE Framework & Implementation

Session 4: Infrastructure

Session 5: Related Standards

Session 6: Platform Users

Session 7: Applications

Session 8: Platform Services

Foreword

As the co-chairs of the DASE Symposium 2001, we would like to welcome you to this second symposium in this continuing series. Once again we have the pleasure of holding the DASE Symposium 2001 at the National Institute of Standards and Technology just outside Washington, D.C., our nation's capital.

The emergence of interactive digital television (DTV) brings about a host of exciting opportunities for broadcasters, content providers, tool developers, and equipment manufacturers. Interactive DTV combines aspects of traditional television and the Internet that inspires applications in e-commerce, e-learning, targeted advertising, video-on-demand, and enhanced viewing services. As the various standards organizations and consortiums hone their lower layer standards for interactive Digital TV, we find that although they derive from common roots they are evolving along different lines. The results are similar but non-interoperable standards for different technologies and different regions. Because of this, interest has shifted to the newly emerging middleware layer standards being developed by the Advanced Television Systems Committee (ATSC) Digital TV Application Software Environment (DASE) specialist group in the USA and Digital Video Broadcast (DVB) Media Home Platform (MHP) in Europe as both an enabling and unifying technology to obtain standardized interactive Digital TV content and behavior. As these middleware layer standards traverse the balloting and acceptance process, it is important to provide more detailed information on their structure, anticipated use scenarios, possible additional features and potential harmonization. That is the purpose of this symposium series. Although our focus at this symposium is on DASE, we have contributions from DVB, the Cable industry, SMJ7rE and others. We hope that the DASE Symposium 200lwiII help to bring about global harmonization in this middleware layer and that future symposiums in this series will focus more broadly on this generic layer.

We hasten to mention that although significant work has been accomplish in the DASE consortium and the structure of the standard is fairly mature, it is important to note that the standard is not finalized and is a work-in-progress.

We would like to thank the speakers for their contributions to this excellent symposium program and, also where applicable, to the DASE effort. We would also like to thank the symposium committee for their hard work and our co-sponsors, ATSC, for their support all of which helped to make this event possible. As most of you already know, putting such a symposium together is an arduous task.

Alan Mink Co-Chair, DASE 2001

Rob Snelick Co-Chair, DASE 2001

Symposium Committee

Co-Chairs

Alan Mink
Rob Snelick
National Institute of Standards and Technology

Publication Chair

Jennifer Quinn
National Institute of Standards and Technology

Exhibit Chair

Christy Kehlbek
Advanced Television Systems Committee

Local Arrangements Chair

Patrice Boulanger
National Institute of Standards and Technology

Registration Chair

Kimberly Snouffer
National Institute of Standards and Technology

DASE Overview, Architecture & Common Content Types

Glenn Adams

XFSI, Inc
glenn@xfsi.com

This talk introduces DASE to newcomers. Basic vocabulary such as 'application' and 'application environment' are defined. General approaches to application deployment are discussed along with key problems. The approach adopted by DASE to these problems is described. A general overview of DASE content and a DASE system is presented along with the status of the draft DASE standard and the schedule for its completion. The expected evolution of DASE as a series of standards is introduced. Outstanding and ongoing problems related to DASE are highlighted.

Building on the introduction, a more detailed review of the DASE content and system architectures are described. The common facilities of both types of application environments are described in some level of detail.

Introduction to DASE

Glenn Adams
Chair ATSC T3/S17

Why DASE?

- The Holy Grail
  - Standardized Interactive Television Content and Behavior
- Useful Side Effects
  - Validate utility and design of A/90 Data Broadcast Standard

General Concepts

- Application
  - Information which expresses behavior
  - A program or a document
- Application Environment
  - System which interprets application in order to produce behavior
  - A program or document processing system

Application Approaches

- Embedded Approach
- Thin-Client Approach
- Full-Client Approach

Embedded Application Approach

- Application pre-installed on receiver
- Generally non-portable; requires re-implementing or porting for new receivers or new technology
- Hard to change or innovate with new applications
- Very stable, but only simple features

Thin-Client Approach

- Application shared between server and receiver
- Application is executed or interpreted on server
- Requires low-latency, high-bandwidth, point-to-point communication channel
- Does not scale well

Full-Client Approach

- Application dynamically installed on receiver through broadcast or point-to-point channel
- Application executed or interpreted on receiver
- Requires more resources and greater performance than thin-client approach

Problem #1: Installing Application

- How to install application on receiver?
  - If pre-installed (embedded), then it is difficult to innovate.
  - If dynamically installed, then application must be transmitted (downloaded) to receiver and prepared for processing in sufficient time for it to be ready to process at the intended time.

Problem #2: Application Form

- What form should an application take?
  - Form = Content Type(s)
- If procedural, then what type?
  - native compiled code
  - portable byte code (p-code)
  - source code
- If declarative, then what type?
  - HTML, XHTML, SMIL, SVG, XML, MHEG
  - graphics, fonts, ...

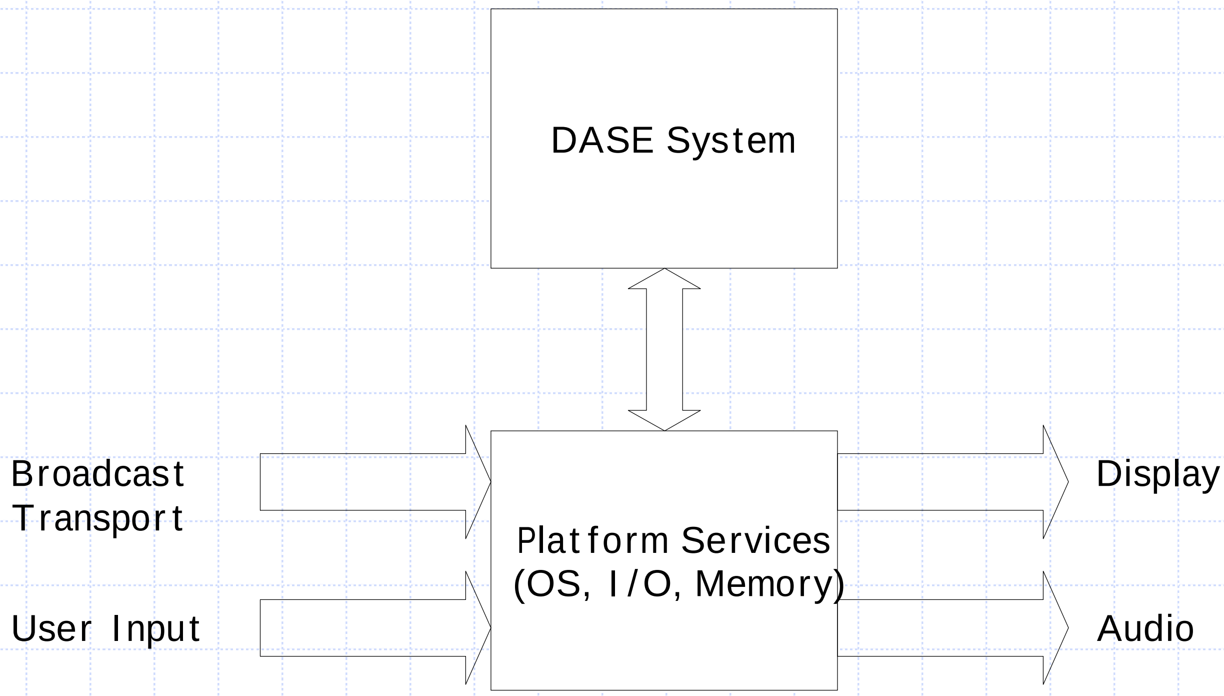
Problem #3: Environment

- What “native” resources may an application reference or utilize?
  - graphics, video, audio, user input (remote/keyboard), broadcast stream, network, memory, processor
- How to reference or use?
  - If mechanism is proprietary, then portability of applications cannot be maintained.

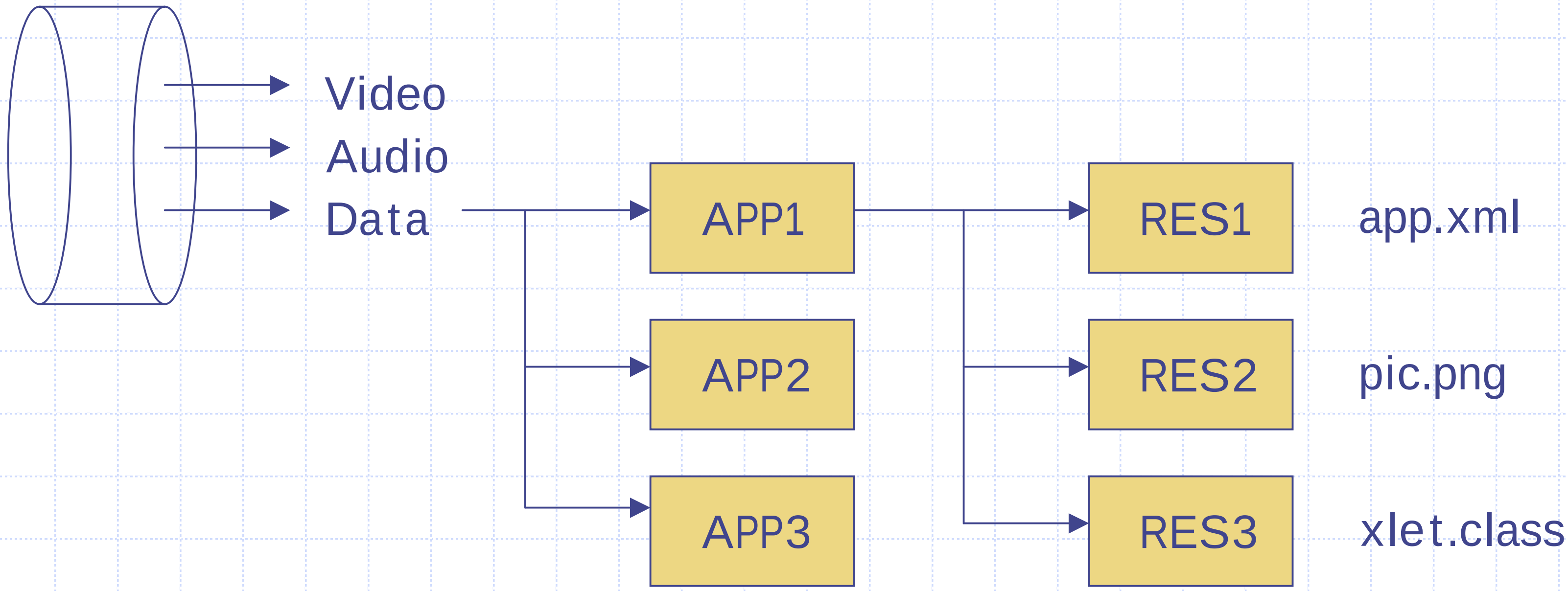
DASE Approach to Problems

- Problem #1: Installing
  - Download through broadcast stream.
- Problem #2: Application Form
  - Standardized form; strict conformance.
- Problem #3: Environment
  - Standardized environment; compliance.

DASE System Interconnect



Application Resources



DASE Application Types

- Declarative Application
- Procedural Application
- Hybrid Application

DASE Declarative Applications

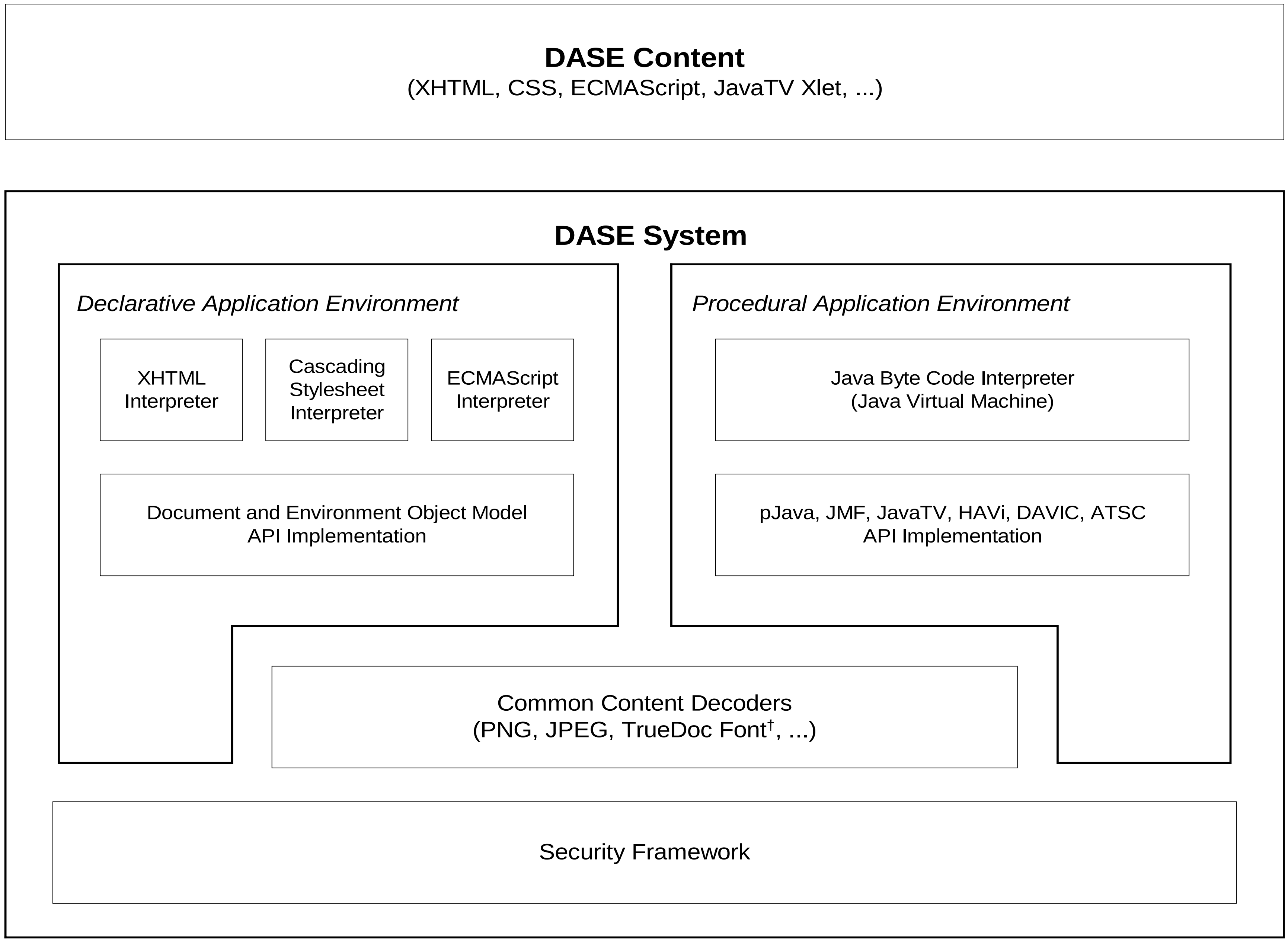
- Declarative Content Type
  - XDML (XHTML Subset)
- Supporting Content Types
  - CSS, ECMAScript, Graphics, etc.
- Document Object Model (DOM)
- Declarative Application Environment
  - System Behavior

DASE Procedural Applications

- Procedural Content Type
  - Java™ Class File Format
- Supporting Content Types
  - Graphics, Audio, Video, etc.
- Procedural Application Environment
  - Java™ Virtual Machine
  - APIs (PJAE, JMF, JavaTV™, HAVi UI, ATSC)
  - System Behavior

DASE Hybrid Applications

- Hybrid Applications
  - Declarative Using Procedural Content
    - Embedded Active Object Content (Xlets)
  - Procedural Using Declarative Content
    - Synthesize Markup, Style, Script Content


DASE Content
(XHTML, CSS, ECMAScript, JavaTV Xlet, ...)
DASE System
Declarative Application Environment
XHTML Interpreter
Cascading Stylesheet Interpreter
ECMAScript Interpreter
Document and Environment Object Model API Implementation
Procedural Application Environment
Java Byte Code Interpreter (Java Virtual Machine)
pJava, JMF, JavaTV, HAVi, DAVIC, ATSC API Implementation
Common Content Decoders
(PNG, JPEG, TrueDoc Font†, ...)
Security Framework

DASE Levels

- Level 1 – Local Interaction
  - Enhanced TV
- Level 2 – Remote Interaction
  - Interactive TV
- Level 3 – Internet Enabled
  - Internet TV

DASE Level One

- Basic Foundation
- Broadcast Only
- No Return Channel

Example DASE-1 Applications

- Play Along Games
  - Jeopardy
- *For More Info*
  - Sport Stats, Product Info (e.g., local car dealer based on user's Zip Code), Local Weather and Traffic Updates
- Mini Program Guide

DASE Level Two

- Builds upon Level One
- Return Channel
- Enhanced Security Framework
  - Digital Signatures
  - Return Channel Encryption
- Plug-Ins, Persistent Applications
- T-Commerce Applications

Example DASE-2 Applications

- Community Gaming
  - Play Against Community of Players
  - Gambling (where legal)
- “T-Commerce”
  - Instant Purchase
  - Coupon Printing
- Full Program Guide

DASE Level Three

- Builds upon Level Two
- General Internet Content
  - Must handle invalid, non well-formed content to be interoperable
- Web TV

Example DASE-3 Applications

- Internet Browsing
  - General Web Access
- Internet Commerce
  - Banking
  - Investment Management
  - C2B, B2B

DASE Standard (1)

- 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

DASE Standard (2)

- Part 6: Security
- Part 7: Application Delivery
- Part 8: Conformance

DASE Development Schedule

- DASE-1 expected to be completed by end of 2001
- DASE-2 requirements development under way

Deployment Challenges

- End-to-End Issues
  - Metadata
  - Format Conversion
  - Synchronization
- Interoperability
  - Conformance Requirements
  - Compliance Testing

Distribution Issues

- Authoring Standard
  - Will authors create native DASE content format or other content to be transcoded into DASE format?
  - SMPTE DDE-2
- Redistribution
  - Will non-terrestrial media (cable and satellite) distribute DASE content?

Harmonization Issues

- DASE, MHP, and OCAP
  - Common declarative functionality
  - Common procedural functionality
  - Many other details differ; but general approach and technology choices are identical.
  - Significant transport level differences

We Need Your Help

- Development of DASE Standard depends upon volunteer commitments.
- Much more work is need to obtain all of the promise of standardized iTV.

DASE Declarative Applications & Environment

Glenn Adams

XFSI, Inc
glenn@xfsi.com

This talk focuses upon the declarative application content and system provided by DASE. The key W3C standards adopted by DASE are reviewed and their use in DASE is described in some detail.

DASE Declarative Applications

Glenn Adams
Chair ATSC T3/S17

Outline

- Declarative Applications
  - Pure Application
  - Hybrid Application
- W3C Technology Usage in DASE
  - Core Technologies and Applications
  - Related Technologies (CSS, DOM)

DASE DA – Pure

- Declarative Application (DA)
  - markup, stylesheet, script content
  - common content types
  - security content types

DASE DA – Hybrid

- Hybrid Applications
  - Declarative Using Procedural Content
    - Embedded Active Object Content (Xlets)
      - Supported in DASE-1

Declarative Content Types

- Markup Content
  - application/xdml+xml
- Stylesheet Content
  - text/css
- Script Content
  - text/ecmascript

Primary Content Types

- XHTML Family Document Type
  - XHTML DTD Driver
  - XHTML Content Model
- Stylesheet Support
  - Cascading Stylesheet (CSS) Level 2 Subset
  - Default Stylesheet
- Scripting Support
  - ECMAScript
  - Document Object Model

XML

- What is it?
  - Extensible Markup Language
  - SGML Subset
- Why do it?
  - SGML overly complex
  - SGML feature abuse leads to poor practice
  - Strong parser requirements: more robust
- How to use it?
  - Define a Document Type Definition (DTD)
  - Create and Validate Document Instance(s)

XML Technologies Used in DASE

- XML 1.0
- XML Namespaces
- XML Stylesheet Linkage
- XML Base
- XML Canonicalization

Related W3C Technologies Used

- Cascading Style Sheet, Level 2
  - Both subsetted and extended
- Document Object Model, Level 2
  - Both subsetted and extended
  - Both ECMAScript (3rd Edition) and Java Bindings

XHTML

- What is XHTML?
  - HTML expressed as XML, not SGML
- But which HTML?
  - XTHML 1.0 (HTML 4.0)
  - XHTML 1.1 (HTML 4.0 subset plus Ruby)
  - XHTML Basic (~HTML 3.2)

XHTML Modularization

- What is it?
  - Division of XHTML 1.0 DTD into Modules
  - Modules define Entities, Elements, Element Content Models and Attributes
- Why do it?
  - Improves reusability; supports customization (subsets, supersets)
- How to use it?
  - Select Modules and Content Models

XDML

- XDML
  - Extensible DTV Markup Language
  - Defines application/xdml+xml content type
- DTD Driver Selects Modules
  - Select modules which provide orthogonal core functionality
- Content Model Entities
  - Defines what can appear as the content of each element type

XHTML Modularization Use

- Two Document Types
  - Standard – Host Language Conformant
  - Frameset – Integration Set Conformant
- Full UA Conformance Not Required
  - Clauses 4-6 Not Required
    - Can reject unknown element, attribute, and attribute value; i.e., can abort if not valid.
- Excludes Certain Modules

Included XHTML Modules

- bidirectional
- client-side image map
- forms
- frame
- hypertext*
- intrinsic events
- list*
- meta
- name identification
- object
- presentation
- scripting
- structure*
- style
- style attribute
- tables
- target
- text*

Excluded XHTML Modules

- applet
  - use <object>
- base
  - use xml:base
- basic forms
- basic tables
- edit
- iframe
- image
  - use <object>
- legacy
  - use style attribute or rule
- link
  - use xml-stylesheet PI
- server side image map

Excluded Element Types

- applet
- base
- basefont
- center
- del
- dir
- font
- iframe
- img
- ins
- isindex
- link
- menu
- s
- u

Stylesheet Support

- Cascading Stylesheets
  - Level 2 Grammar
  - All Level 1 Properties
  - Some Level 2 Properties
  - Some Level 2 Property Value Extensions
  - Some ATSC Specific Extensions
- Default Stylesheet

Using Stylesheets

- External Stylesheet
  - Uses <?xml-stylesheet href="…"?>
- Internal Stylesheet
  - Uses <style>…</style>
- Anonymous Style Rules
  - Uses "style" attribute, e.g., <p style="{color=red}">…</p>

CSS2 Subset (1)

- Includes all CSS1 properties
- Includes subset of new CSS2 properties
  - border-{bottom,top,left,right}-{color,style}
  - bottom, top, left, right, z-index
  - caption-side
  - clip, overflow
  - content, counter-{increment,reset}
  - outline, outline-{color,style,width}
  - position
  - visibility

CSS2 Subset (2)

- Partial Property Semantics
  - display, font
    - CSS1 values only plus ‘inherit’
  - list-style-type
    - CSS1 values plus ‘decimal-leading-zero’, ‘lower-latin’, ‘upper-latin’, and ‘inherit’
  - text-decoration
    - CSS2 values minus ‘blink’

CSS Subset (3)

- Includes all CSS2 selectors except:
  - adjacent sibling
  - child
  - :first-child pseudo-class
  - :hover pseudo-class
  - :lang pseudo-class

CSS Subset (4)

- Partial @font-face rule semantics
  - 'font-family', 'font-style', 'font-variant', 'font-weight', 'font-stretch', 'font-size' descriptors
  - 'unicode-range' descriptor
  - 'src' descriptor
- Excludes @page rule semantics

CSS2 Extensions

- ‘atsc-tv’ media type
- ‘atsc-rgba(*r*,*g*,*b*,*a*)’ function
- *style* attribute syntax
  - permits inline rulesets
  - based on “Syntax of CSS rules in HTML’s style attribute”, W3C Working Draft, 25 October 2000

Scripting Support

- ECMAScript (ECMA-262)
  - Third Edition (adds exceptions, regexp)
- (Language) Native Objects
  - Global, Object, Function, Array, String, Boolean, Number, Math, Date, RegExp, Error
- Host Objects
  - Document Object Model

Document Object Model

- What is it?
  - A means to create and manipulate a parsed representation of a document instance (e.g., an XDML document)
- Why do it?
  - Content adaptation, Dynamic Style Application, Document Synthesis
- How to use it?
  - Use <script>…</script>
  - Use intrinsic events (e.g., onmouseover)
  - Use triggers

DOM2 Subset

- Excluded Modules
  - CSS2 (CSS2 Extended Interfaces)
  - Range
  - Traversal
- Excluded Interfaces
  - Excludes all HTML Module interfaces except those required for “DOM-0” legacy script content

Included HTML Module Interfaces

- HTMLAnchorElement
- HTMLDocument
- HTMLFormElement
- HTMLOptionElement
- HTMLBodyElement
- HTMLDOMImplementation
- HTMLInputElement
- HTMLSelectElement
- HTMLCollection
- HTMLElement
- HTMLObjectElement
- HTMLTextAreaElement

DOM2 Extensions

- Adds Modules
  - Legacy
- Adds Interfaces
  - Adds interfaces to Core, Views, and HTML modules

DOM2 Core Module Extensions

- DOMExceptionExt
  - VALIDATION_ERR
  - NO_CLOSE_ALLOWED_ERR

DOM2 View Module Extensions

- DocumentViewExt
  - width{Px,Mm}, height{Px,Mm}
  - sampleBits{R,G,B,A}
  - refreshOnChange

DOM2 HTML Module Extensions (1)

- HTMLAnchorElementExt
  - hash, host, hostname, pathname, port, protocol, search
- HTMLDocumentExt
  - location, lastModified, {a,,v}linkColor,{bg,fg}Color, window, clear()
- HTMLFormElementExt
  - encoding
- HTMLObjectElementExt
  - complete, lowsrc, src

DOM2 HTML Module Extensions (2)

- HTMLTriggerObjectElementExt
  - backChannel, contentLevel, sourceId, enabled, releasable

DOM2 Added 'Legacy' Module (1)

- History
  - length, back(), forward(), go()
- Location
  - hash, host, hostname, href, pathname, port, protocol, search
- Navigator
  - appName, appVersion, appCodeName, userAgent, ddeBackChannel, ddeContentLevel, ddeSourceId, ddeEnabled, ddeReleasable

DOM2 Added ‘Legacy’ Module (2)

- Window
  - document, history, location, navigator
  - frames, length, parent, self, top, window
  - defaultStatus, name, opener, status
  - alert(), confirm(), prompt()
  - setTimeout(), clearTimeout()
  - close(), open()

Validity and a Mutable DOM

- What is the problem?
  - DOM permits mutating a valid document in ways that may cause it to become invalid
  - However, a series of DOM mutations, though invalid in intermediate steps, may be valid in the final step
- Current Status
  - Content must insure validity – result is undefined (i.e., implementation dependent) if not valid

DA Design Goals

- Employ best current and emerging practice from W3C
- High syntactic and functional orthogonality
- Support *functionality* of SMPTE Declarative Data Essence, Level 1 (DDE-1) through transcoding

Transcoding Requirements

- Cannot transcode script content due to run-time script content synthesis, e.g., eval()
- Should be able to support one-pass, no look-ahead transcoding of DDE-1 to XDML in order to meet stringent real-time broadcast constraints

Transcoding Problems (1)

- document.write()
  - can produce same effect as eval()
  - can't transcode because may rely upon runtime state to produce results
  - generates DDE-1 content
  - therefore, must perform transcoding on receiver also; i.e., transcode output of document.write() prior to parsing as XDML

Transcoding Problems (2)

- Interaction of ‘name’ attribute and script content precludes reliable translation of ‘name’ into ‘id’, e.g.,

```
<form name=‘foo’>
<input name=‘foo’ value=‘bar’>
</form>
<script>
function consMutator ( fn, en, val )
{
  return “document.” + fn + “.” + en + “.value=” + val;
}
eval ( consMutator ( ‘foo’, ‘foo’, ‘baz’ ) );
</script>
```

Transcoding Problems (3)

- W3C has deprecated functionality without defining alternative, e.g.
  - <frameset>, <frame>
  - <hr noshade>
  - <legend align=“bottom”>

Transcoding Problems (4)

- Style Rule vs Style Attribute
  - Simple transcoding requires translation of presentation attributes into style attributes; however,
  - If style rule already applied to element, then synthesized style attribute may conflict with precedence of style rule.
  - Resolution requires promotion of inline style rule to stylesheet.

DASE API Object Model and Examples of Use

Petr Peterka

Motorola Broadband Communications Sector, San Diego, CA PPeterka@ gi.com

As DASE passed the initial balloting process, it becomes more and more important to provide not only reports on the design progress, updated lists of features, satisfied requirements and the basic structure of the API packages but also the relationships between these APIs, typical use cases and sample applications. There are two primary users of the DASE standard: (1) the DTV receiver implementers who will implement these APIs on their specific devices and (2) the content authors who will be writing applications accessing these APIs without any detailed knowledge of the target device. Our presentation will primarily focus on the needs of the content authors.

Since there are similar efforts in different realms of the TV industry, DASE decided to reuse existing APIs where appropriate. As a result, the DASE specification includes the following APIs: Sun's Java TV 1.0 and JMF 1.1 APIs, HA Vi 1.1 User Interface API, W3C DOM APIs, a subset of DA VIC 1.4 APIs and an ATSC-specific set of APIs. All of these APIs are defined on top of the Java Virtual Machine and a subset of Personal Java 1.2. Personal Java provides the basic Java packages, which abstract an operating system; Java TV provides the core DTV receiver functionality including tuning, access to system and service information, data carousels, extensions to JMF, etc.; HA Vi addresses the needs of an embedded device with respect to a light-weight user interface; the W3C DOM API provides a bridge between the DASE declarative and procedural applications. Finally, DASE adds APIs for ATSC-specific features including PSIP (A65) and the ATSC data broadcast protocol (A90). Other extensions include support for application management, user management and user preferences. An Xlet, a broadcast version of an Applet, represents downloadable applications, which are delivered as data in the MPEG-2 transport stream together with audio, video and supporting data.

The DTV receiver system services that are being abstracted by the Java APIs include Network Communication, Content Management, Presentation and User Interface, Application and Resource Management, Security Management, Environment Management and Utility Services.

The main focus of this presentation is not an exhaustive detailed description of all Java APIs that are included in the DASE standard but rather an overview of the more significant or complex packages with the emphases on their integrations and use cases from the content authoring point of view. We will review the main parts of the object model represented in UML notation and we will use sequence diagrams to show interaction between the Xlet and selected DASE, Java TV, HA Vi ill and Personal Java APIs. It will give content authors an idea of how these APIs can be used together to produce very appealing content. Main examples will include Xlet startup and initialization, access to data carousel files, setting up an IP multicast stream, working with user preferences, accessing the current service information, retrieving a list of channels from the PSIP database and browsing a program schedule on a given service.

intelligence everywhere™

DASE Symposium 2001

DASE API Object Model & Examples of Use

Petr Peterka

Motorola Broadband Communications Sector

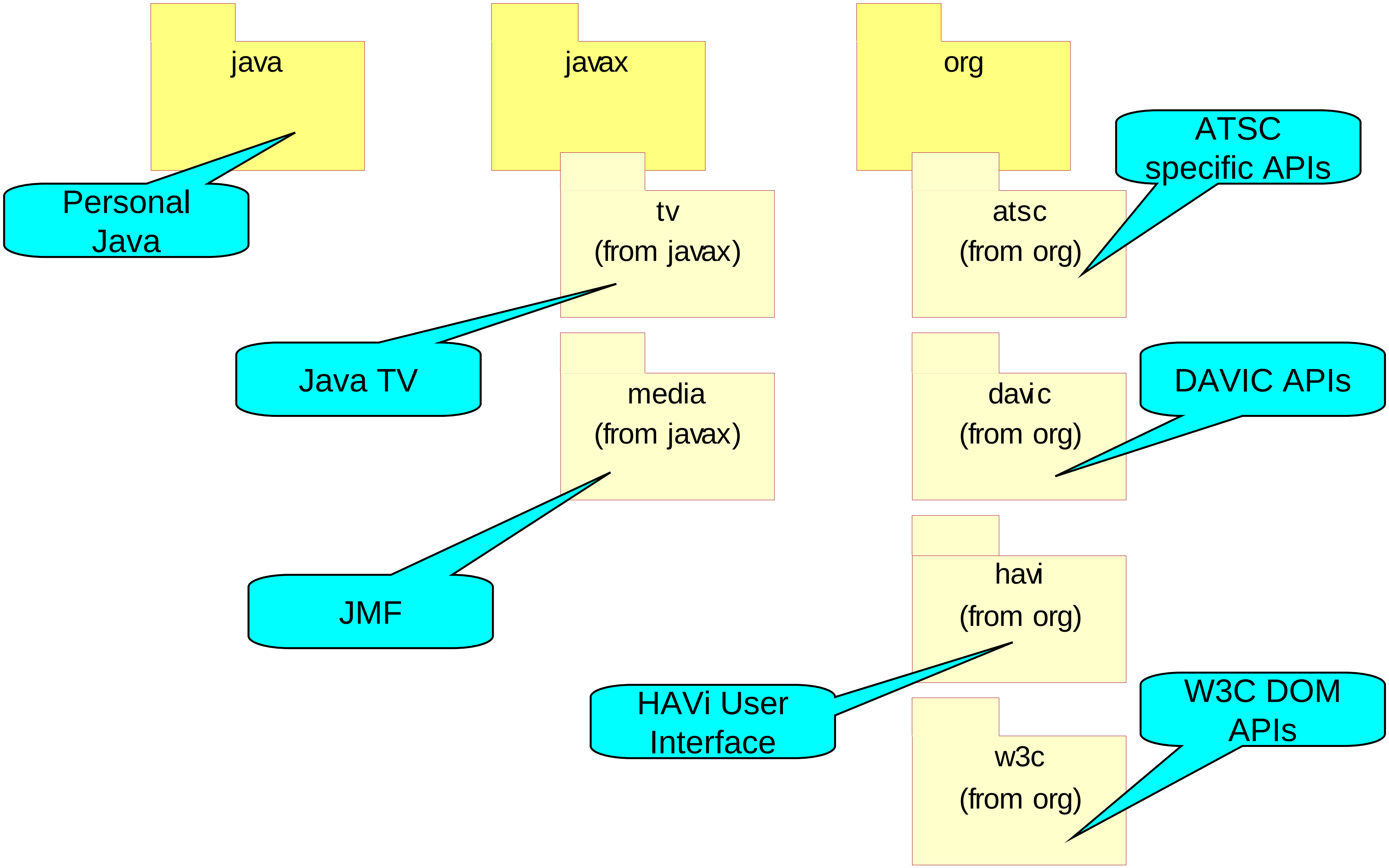
19 June, 2001

Contents

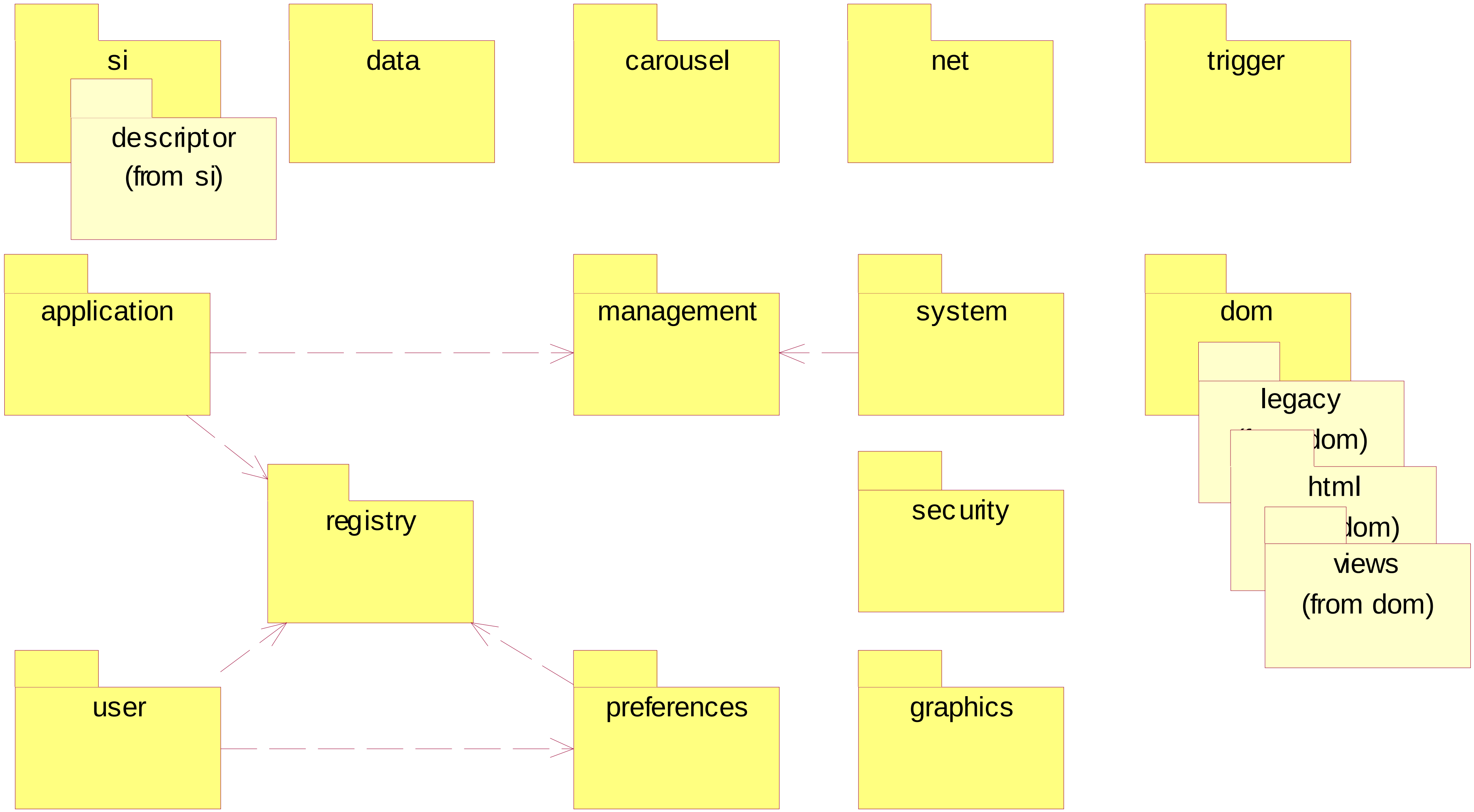
Design Process

- Sample Applications
- Definition of System Services
- Reduced Set of DASE-1 Requirements
- Search for Existing APIs
- Call for Proposals
- Selection and refinement
- Mapping against Requirements
- Subsetting and Constraining

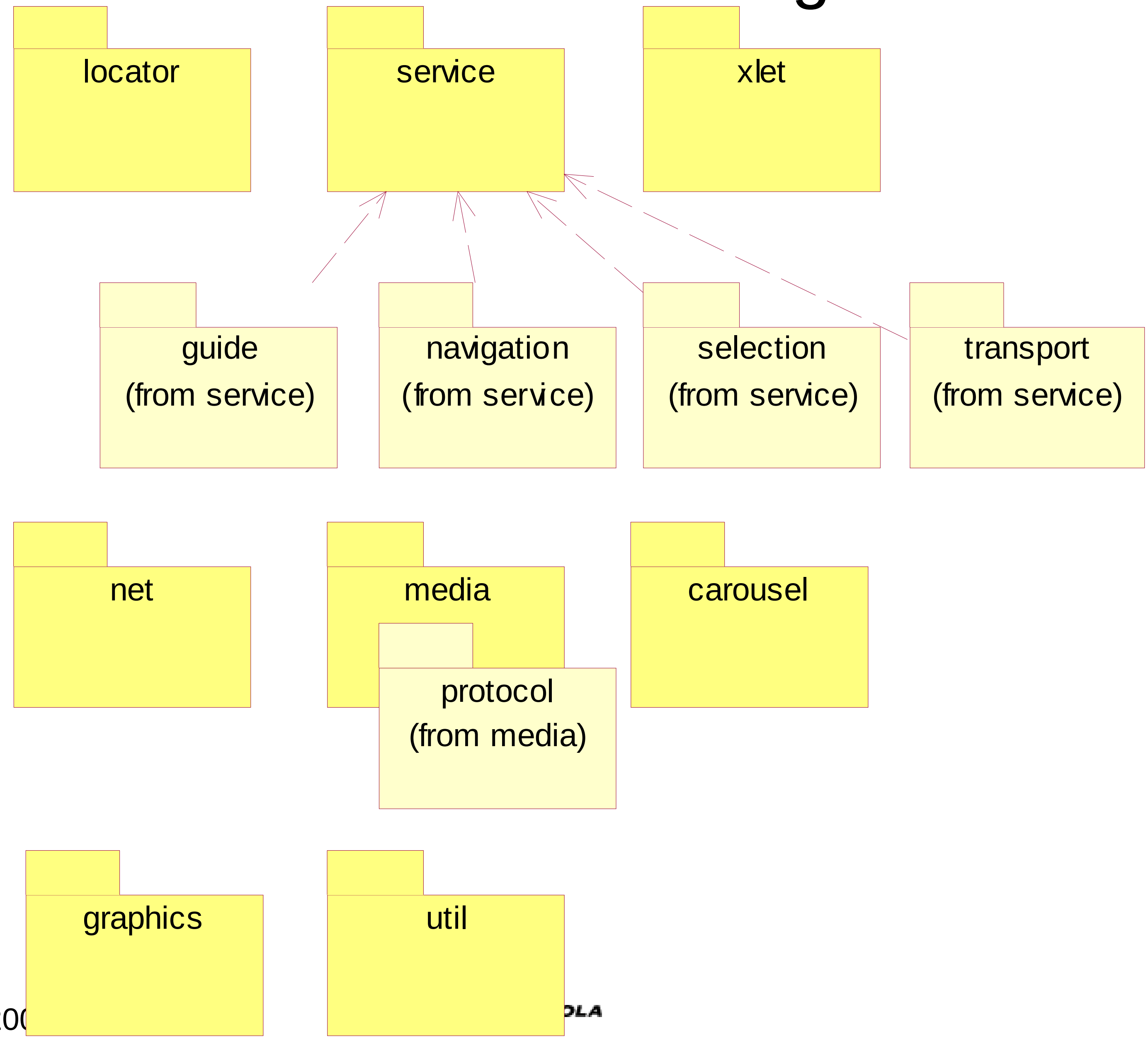
Java Package Structure



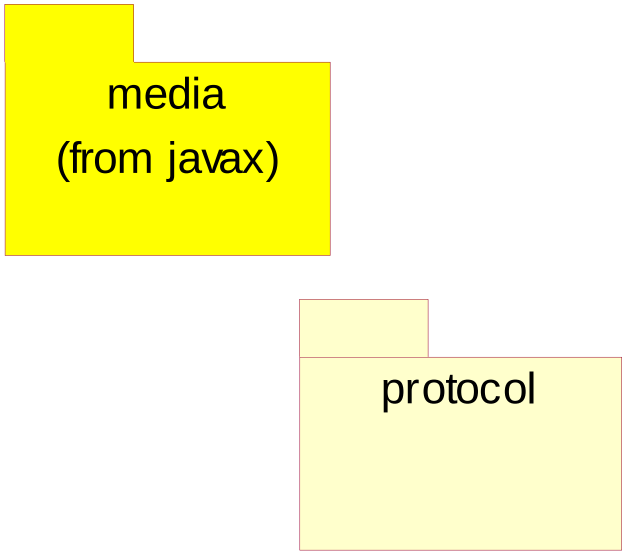
DASE Packages



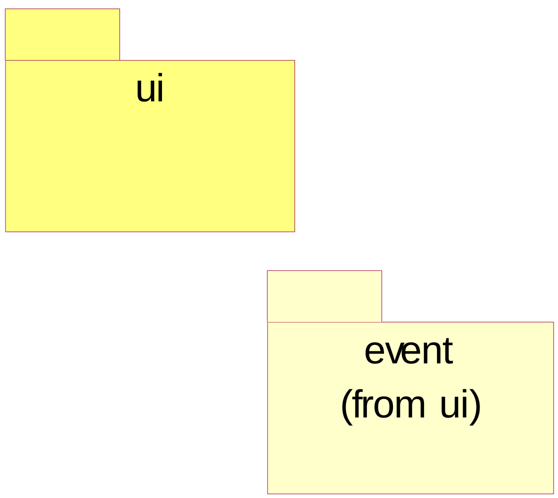
Java TV^{tm} Packages



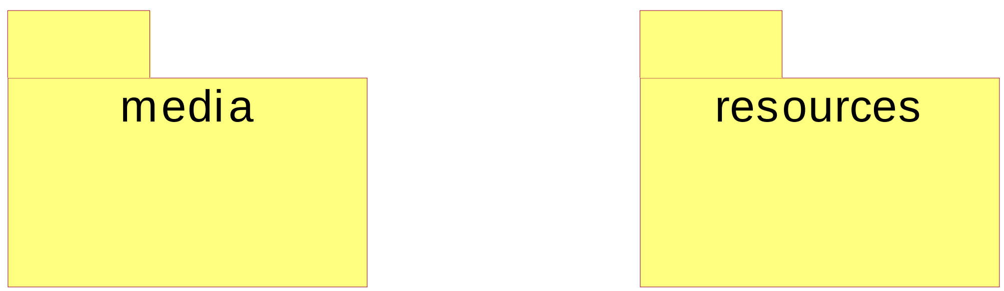
JMF Packages



HAVi UI Packages

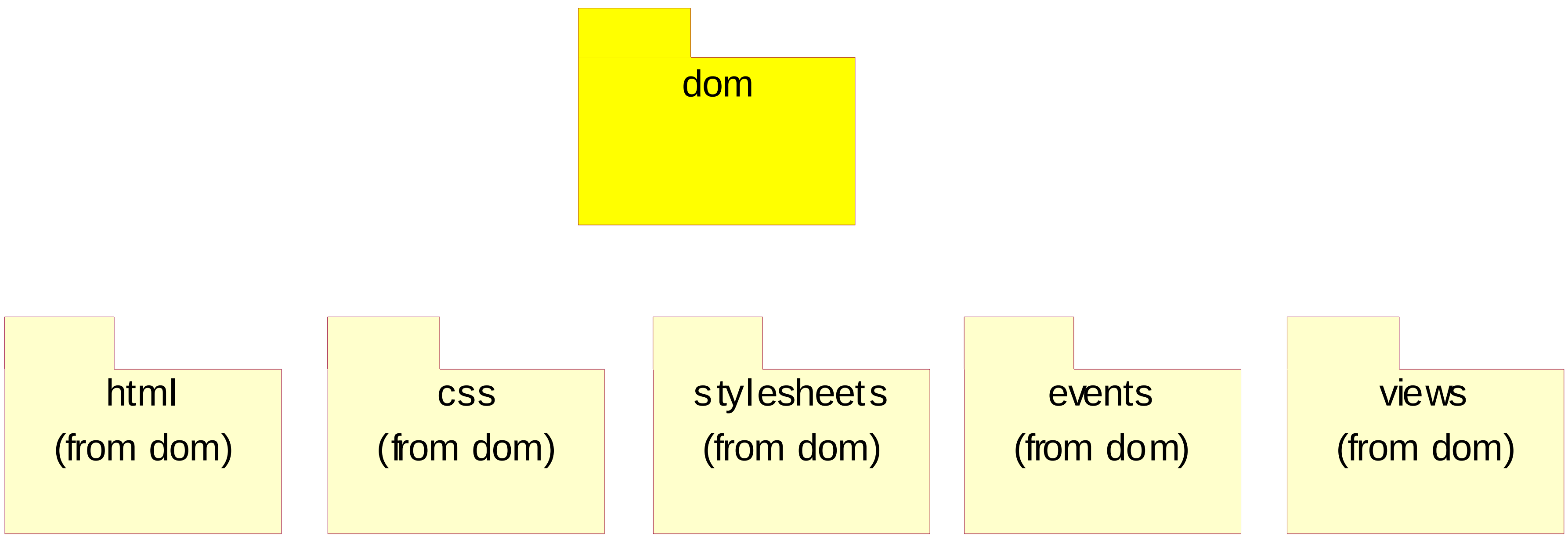


DAVIC Packages

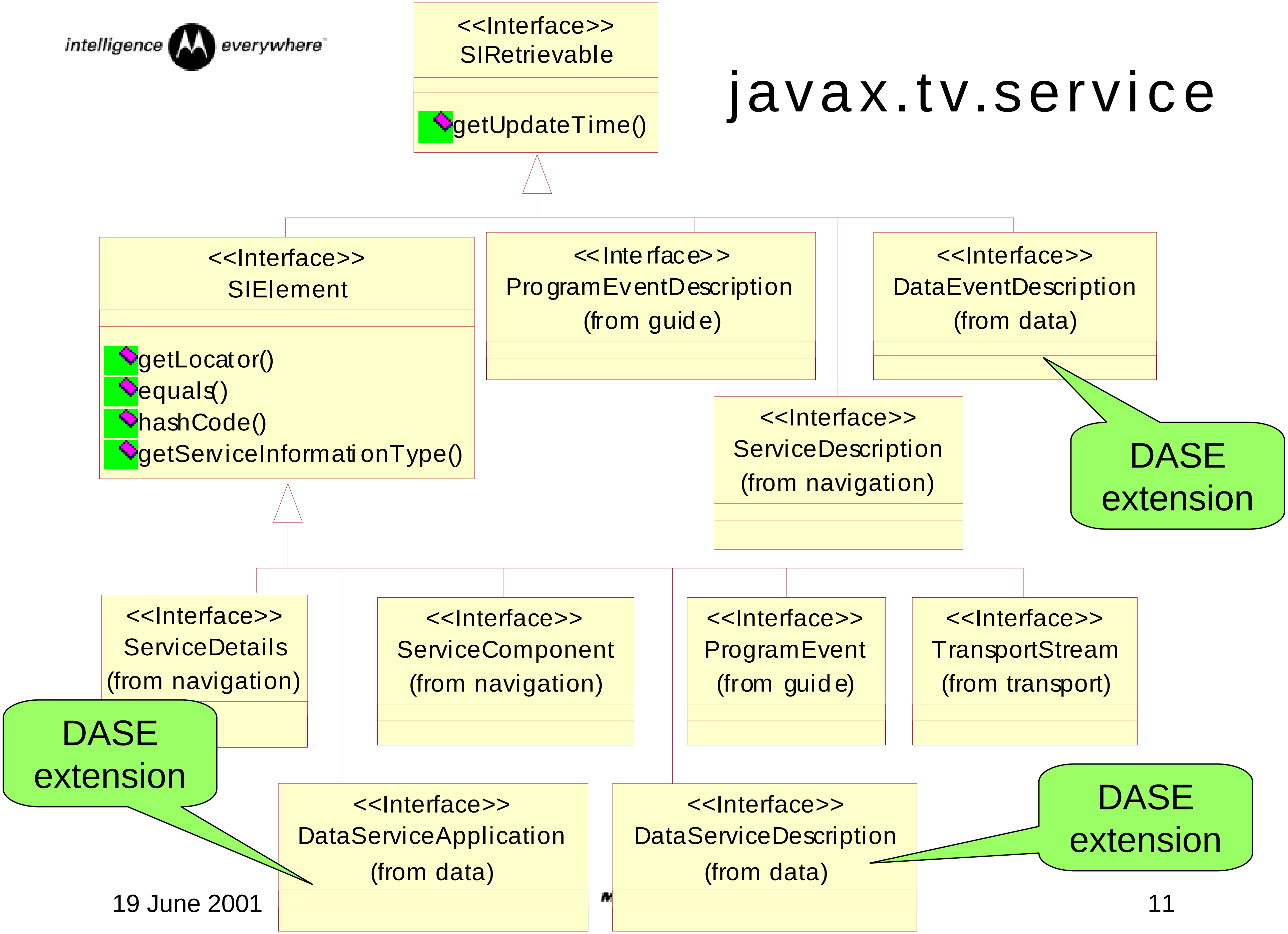


intelligence everywhere

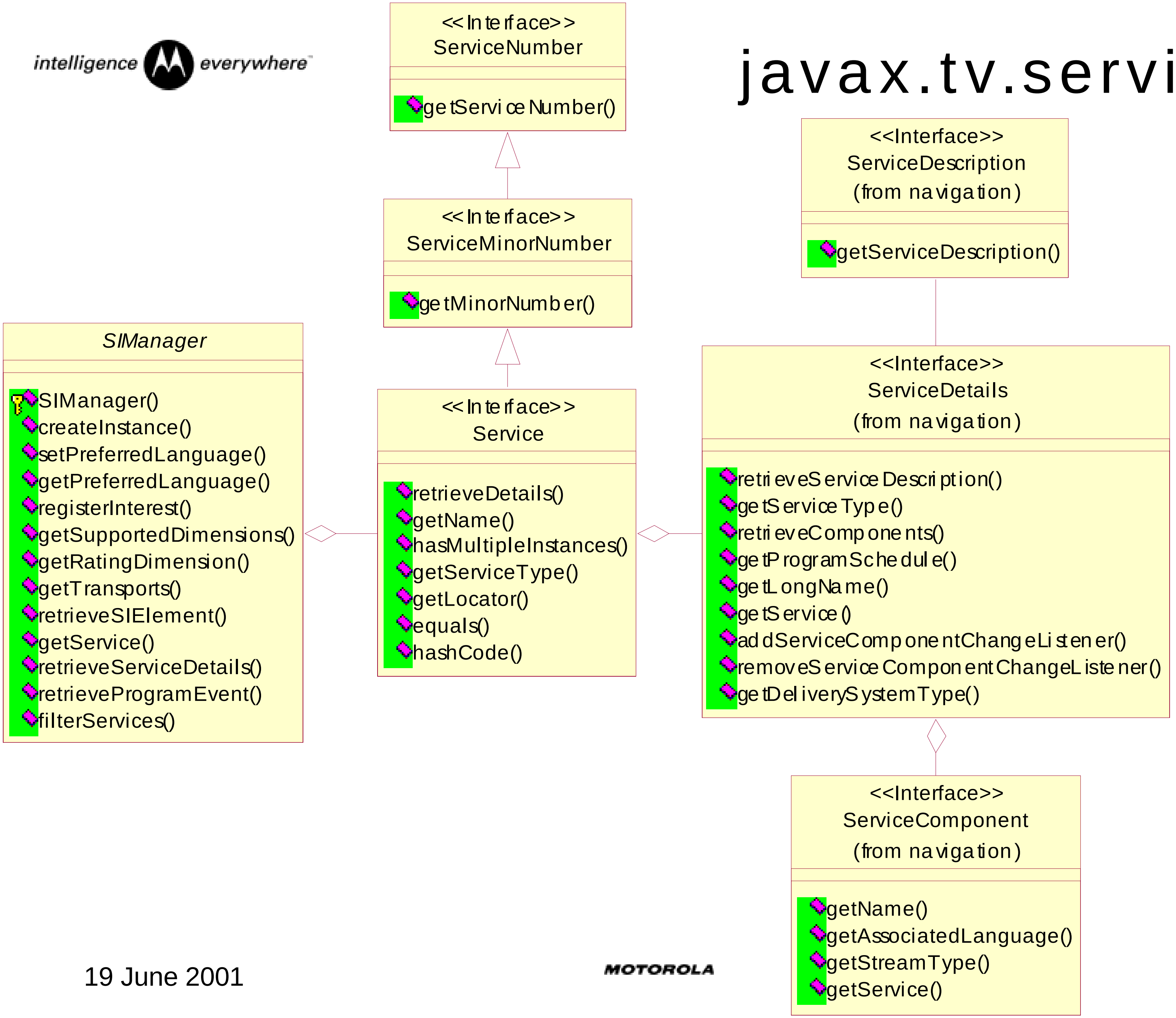
W3C DOM Packages



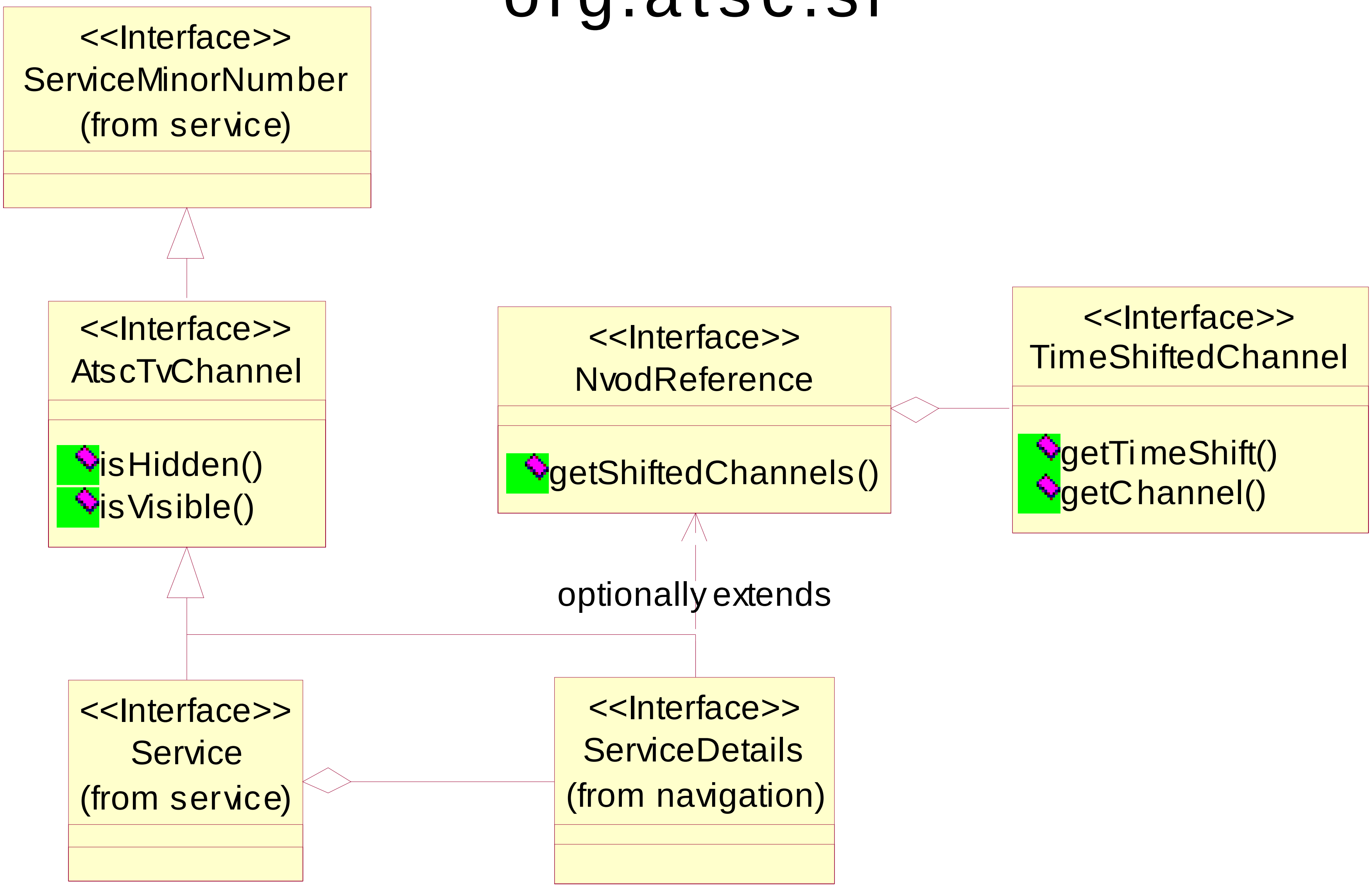
javax.tv.service



javax.tv.service

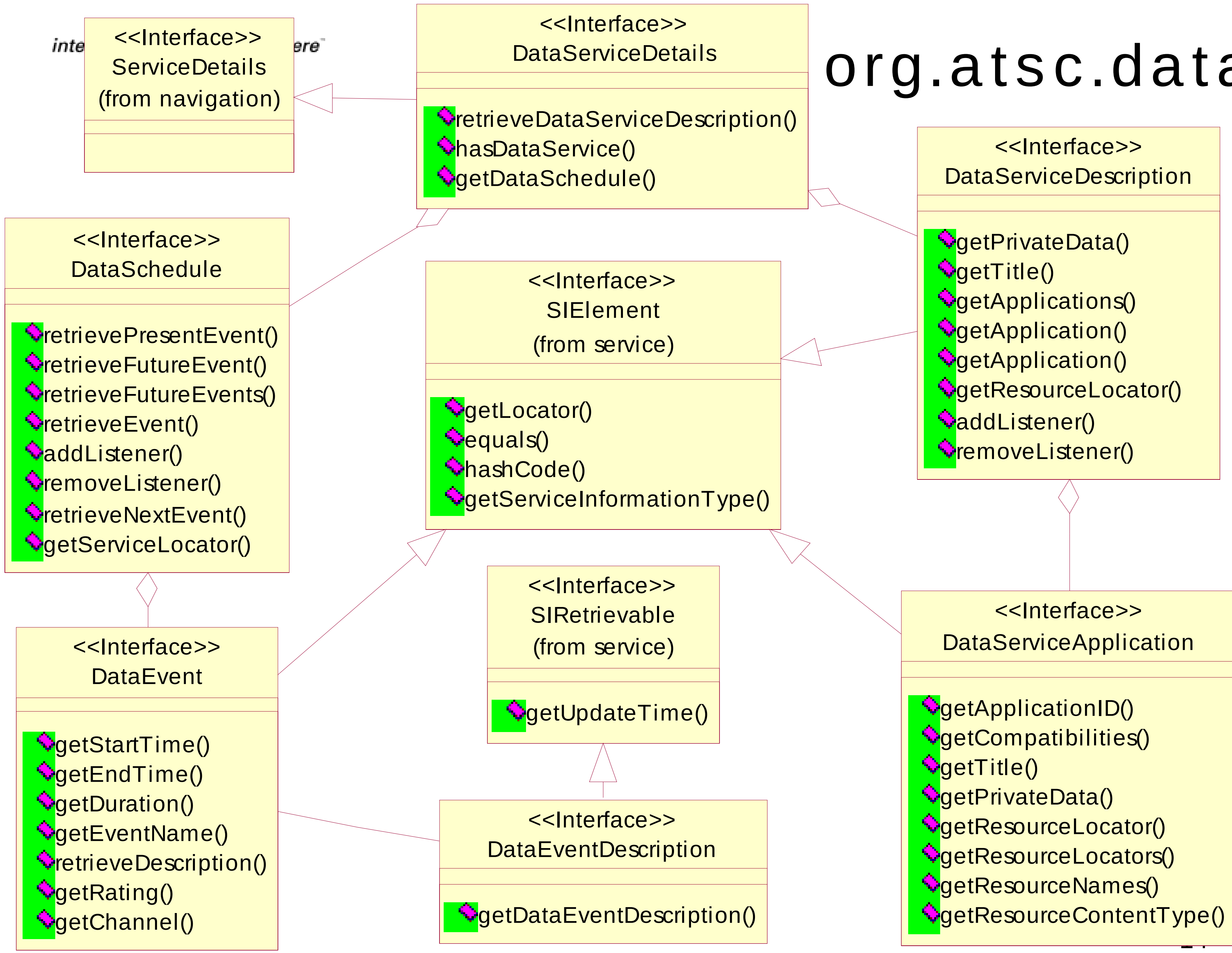


org.atsc.si

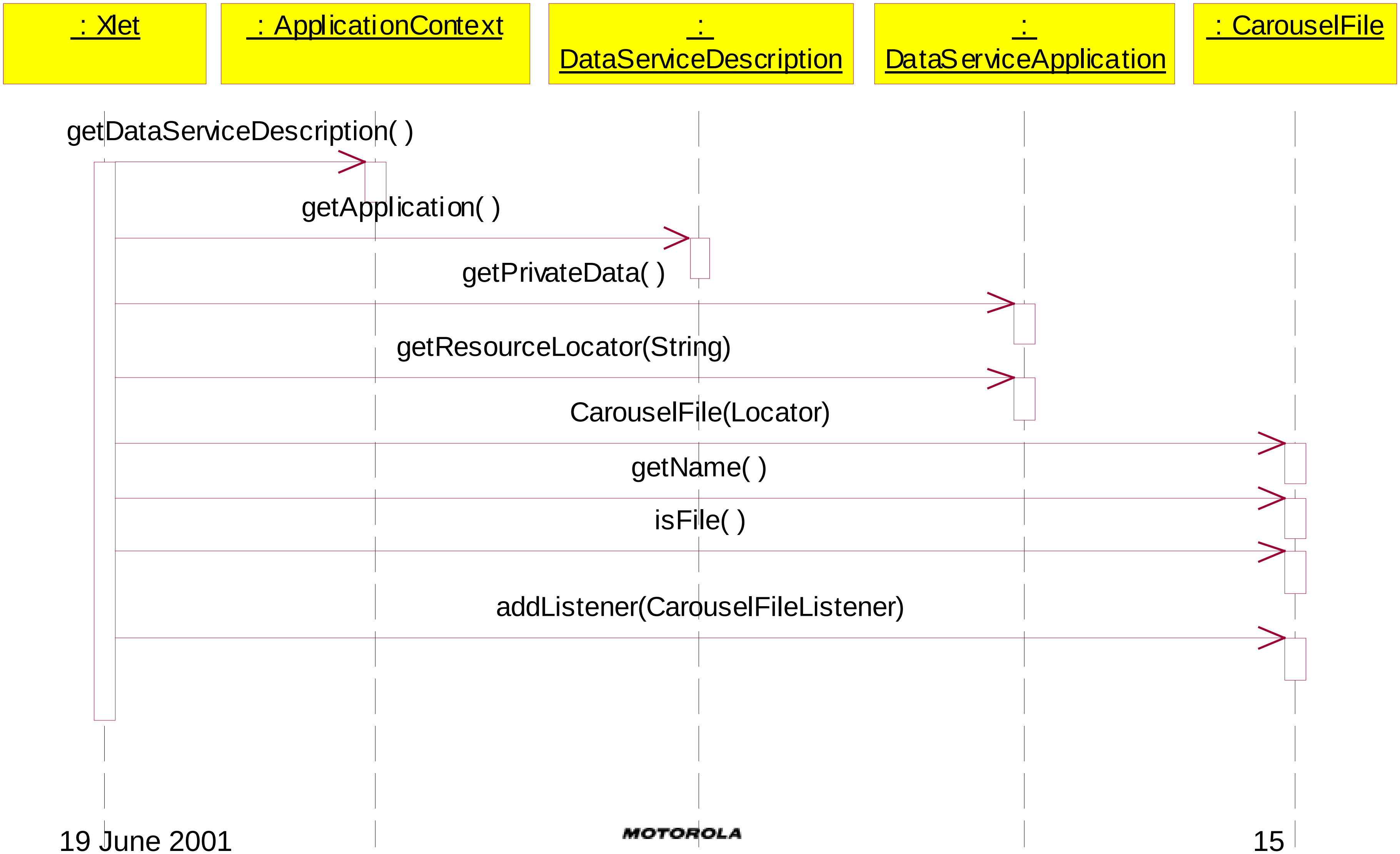


<<Interface>>
ServiceMinorNumber
(from service)
<<Interface>>
AtscTvChannel
isHidden()
isVisible()
<<Interface>>
NvodReference
getShiftedChannels()
<<Interface>>
TimeShiftedChannel
getTimeShift()
getChannel()
optionally extends
<<Interface>>
Service
(from service)
<<Interface>>
ServiceDetails
(from navigation)

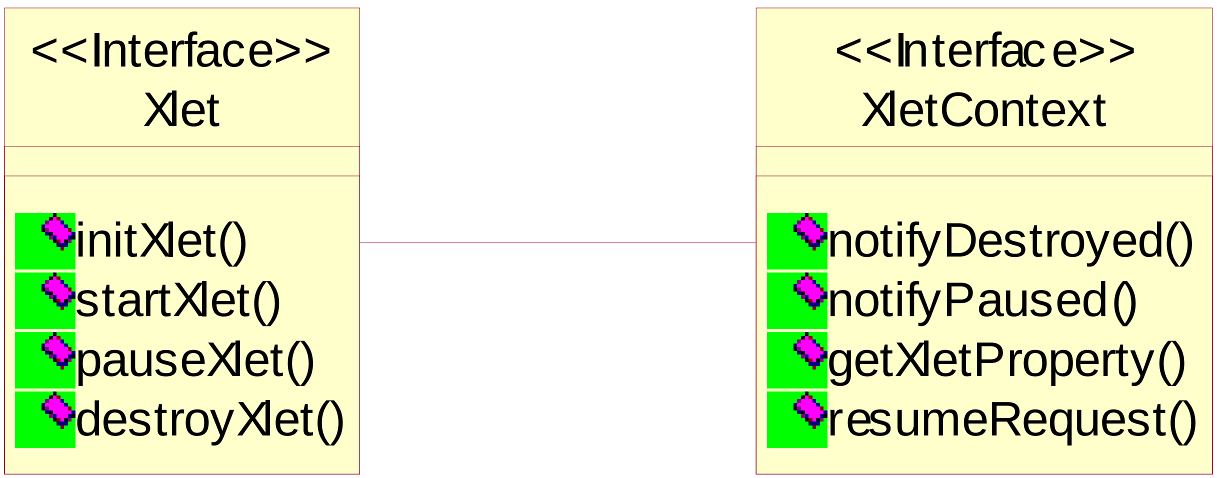
org.atsc.data



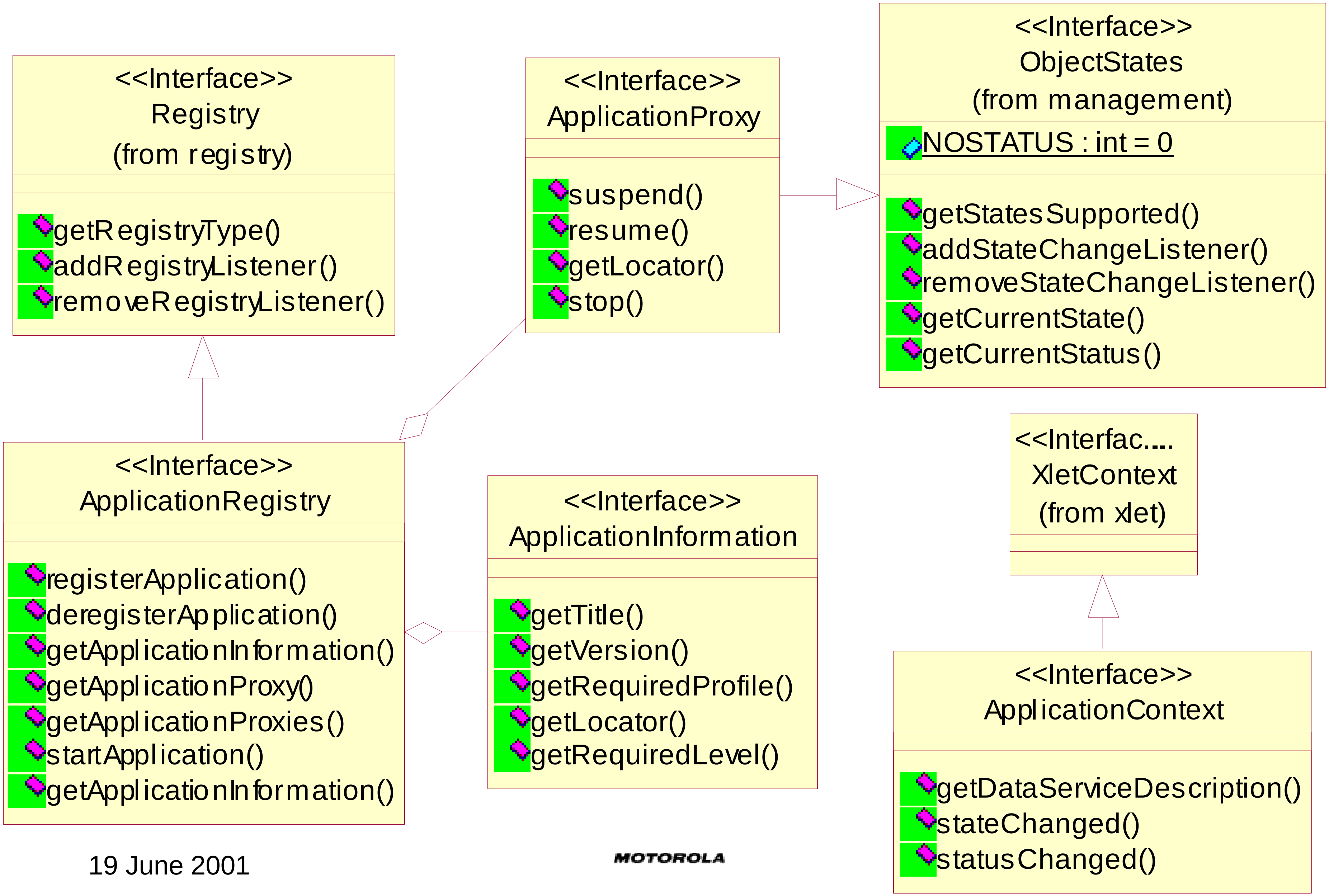
Data Carousel Scenario



javax.tv.xlet

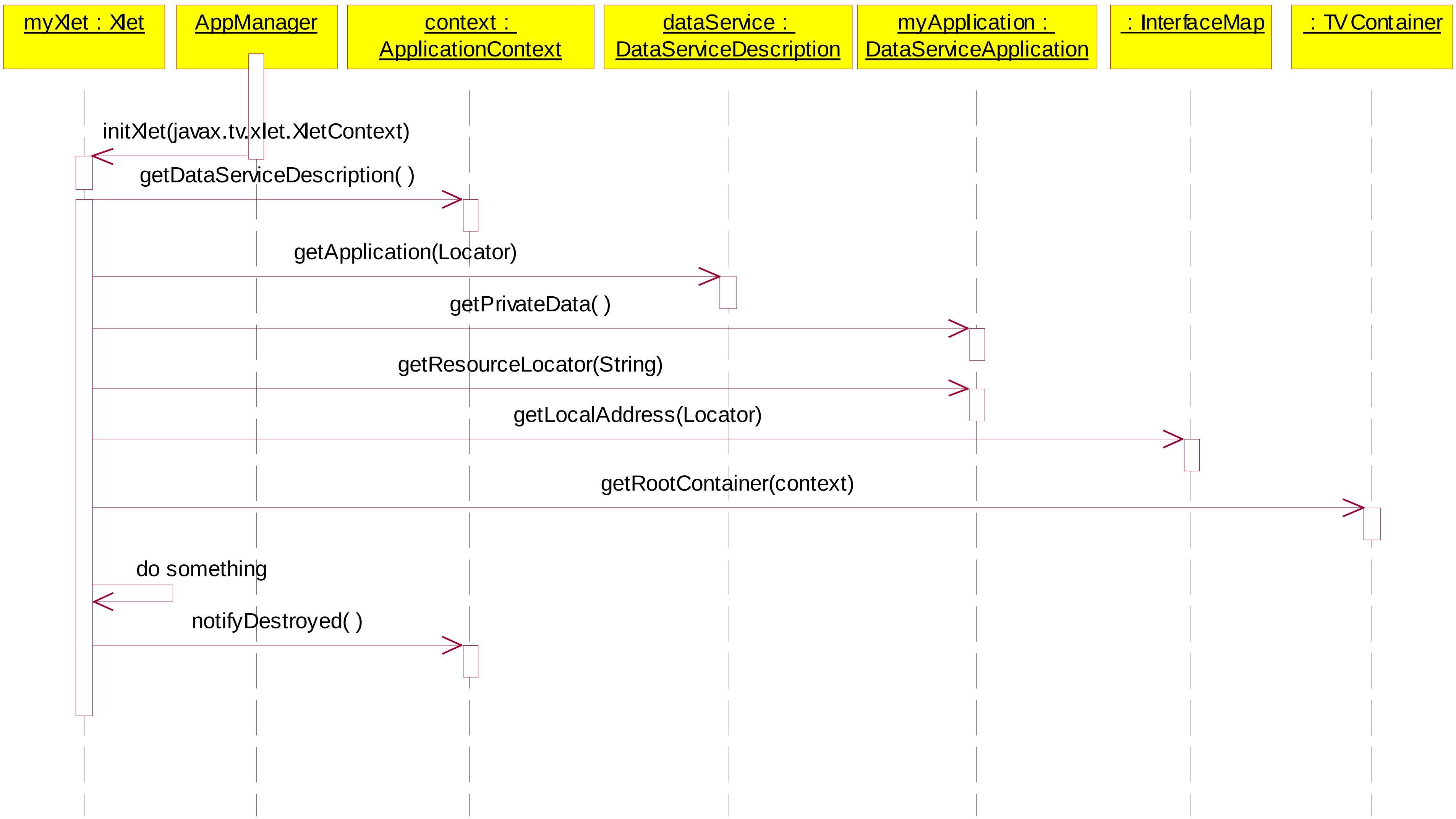


org.atsc.application

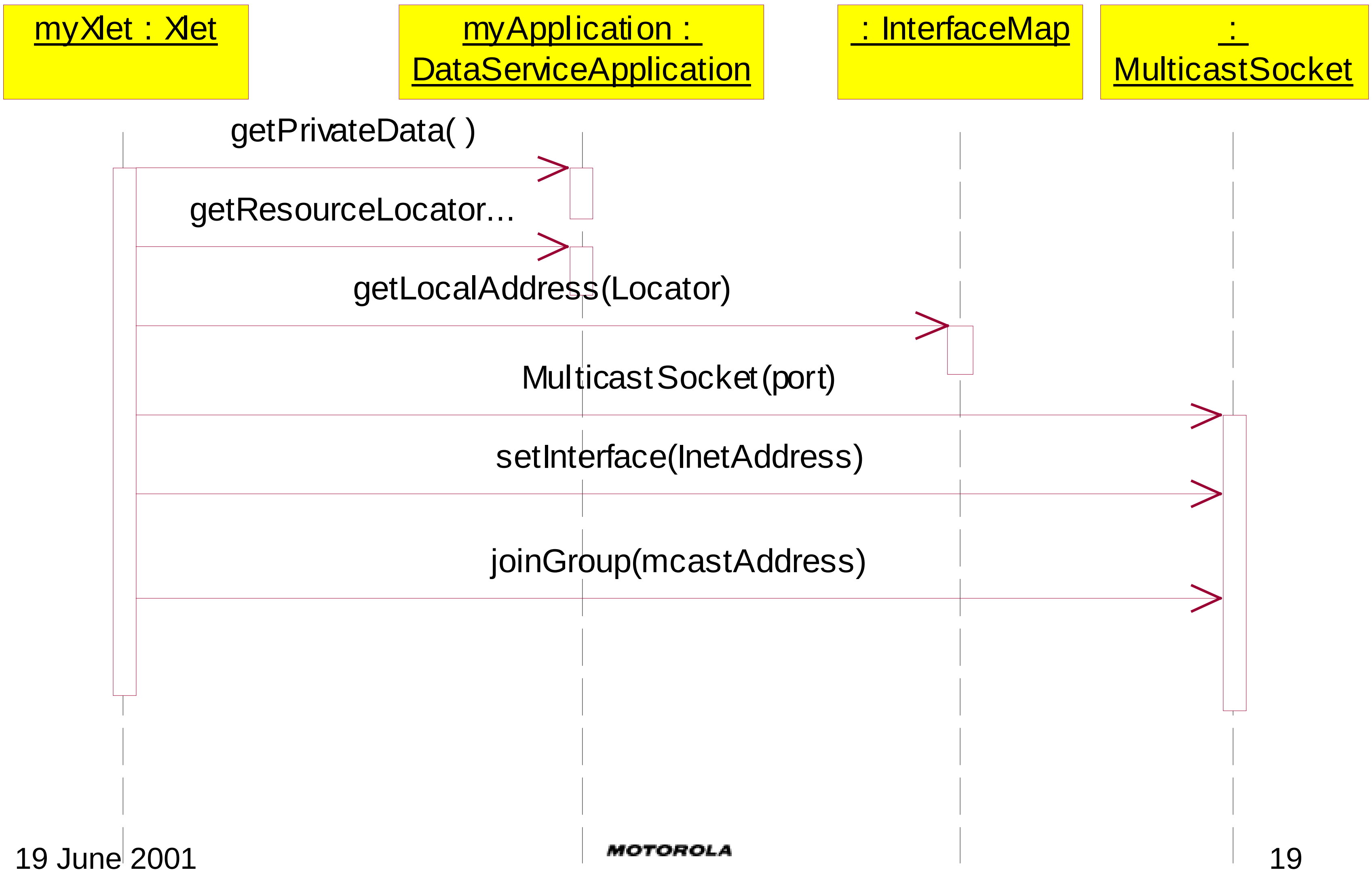


intelligence everywhere

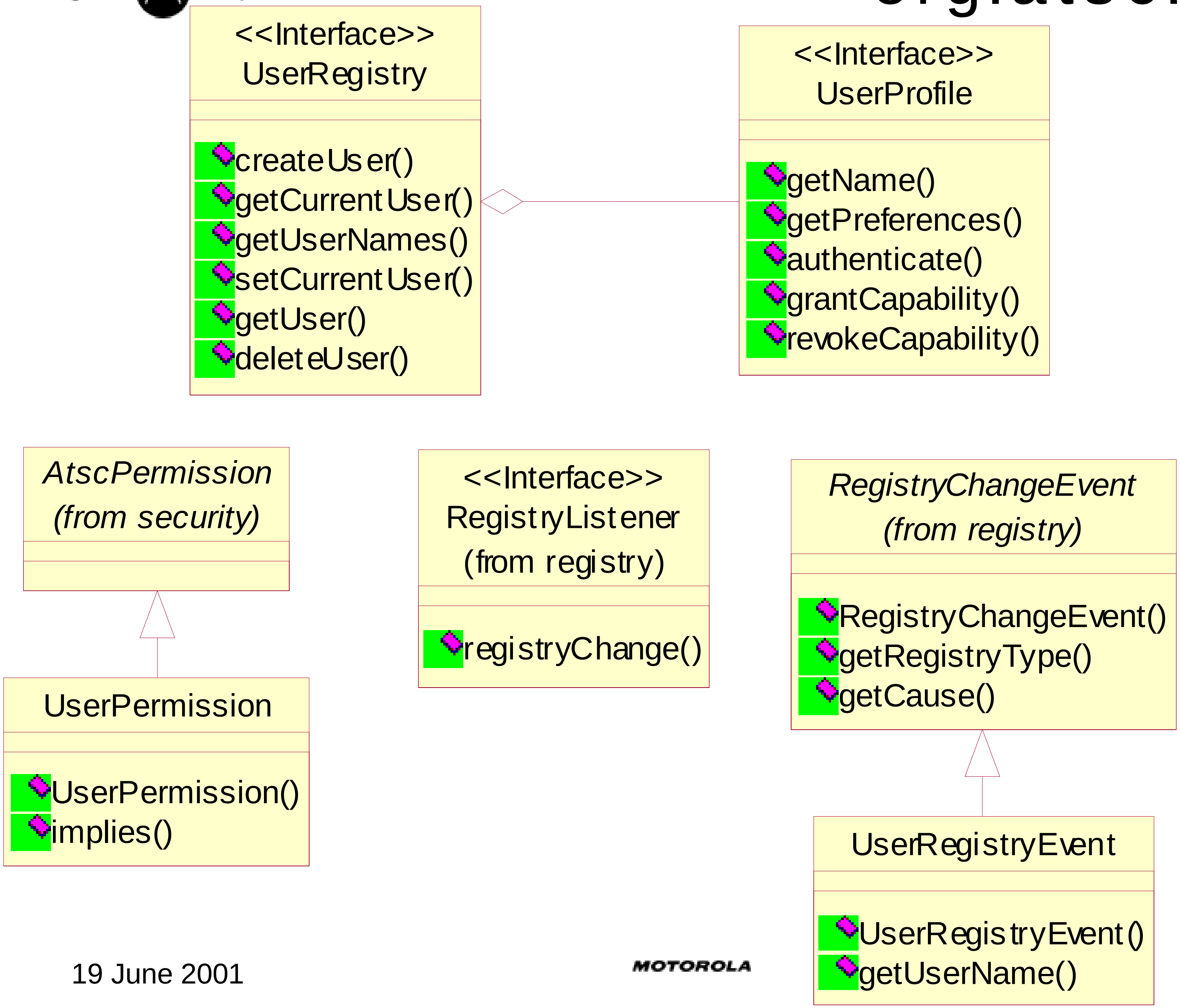
Xlet Start-up Scenario



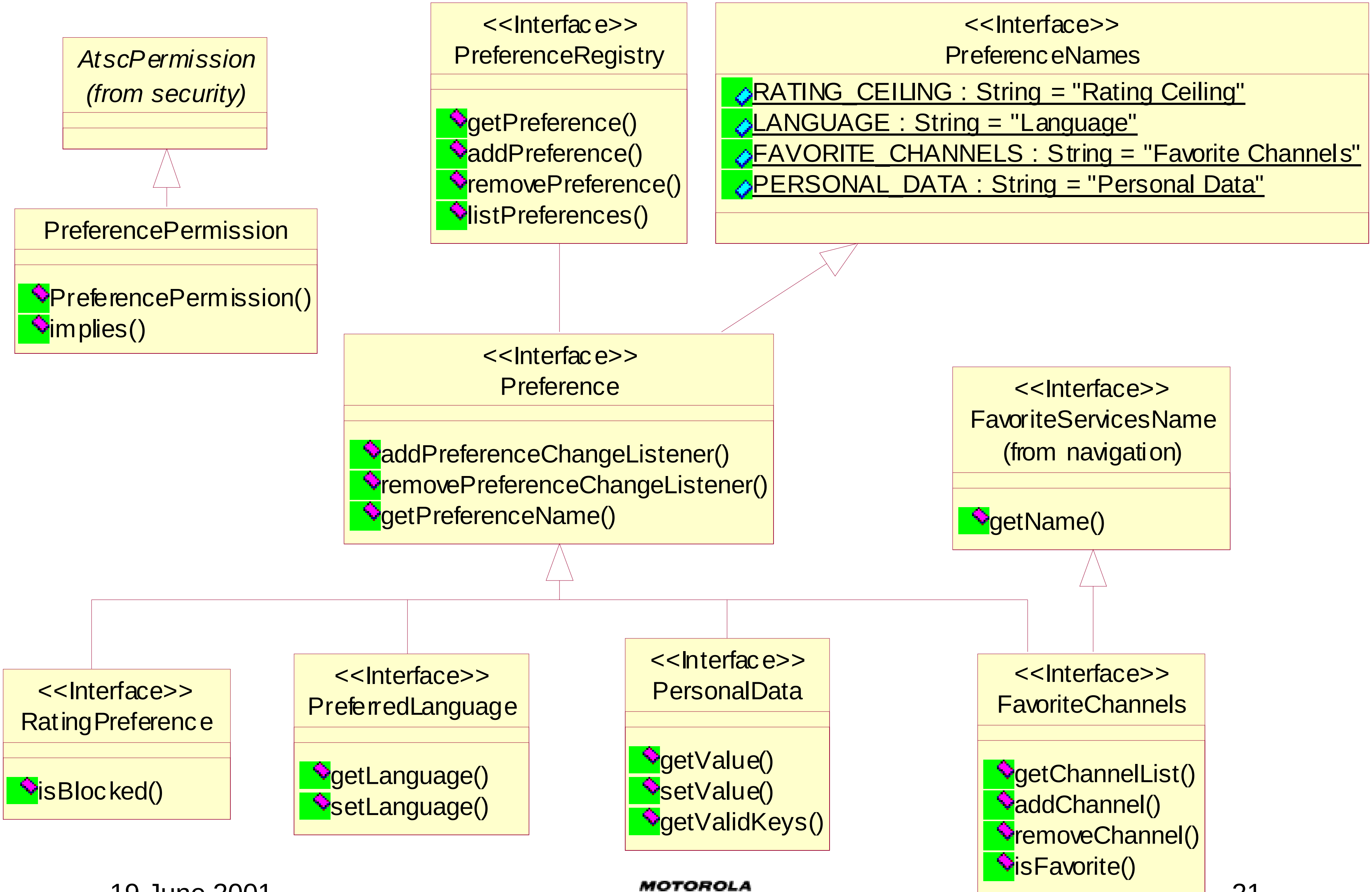
Receiving IP Multicast



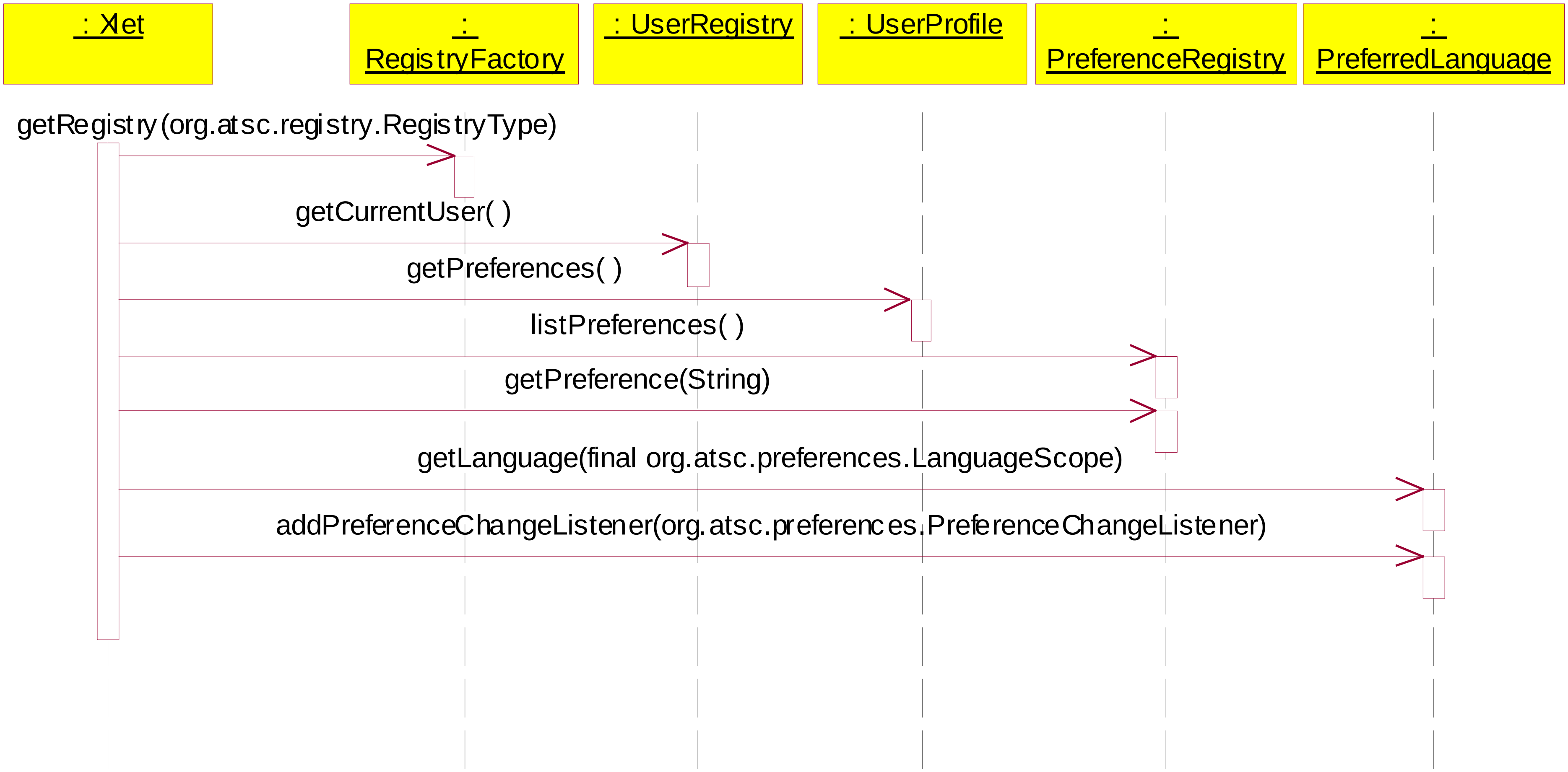
org.atsc.user



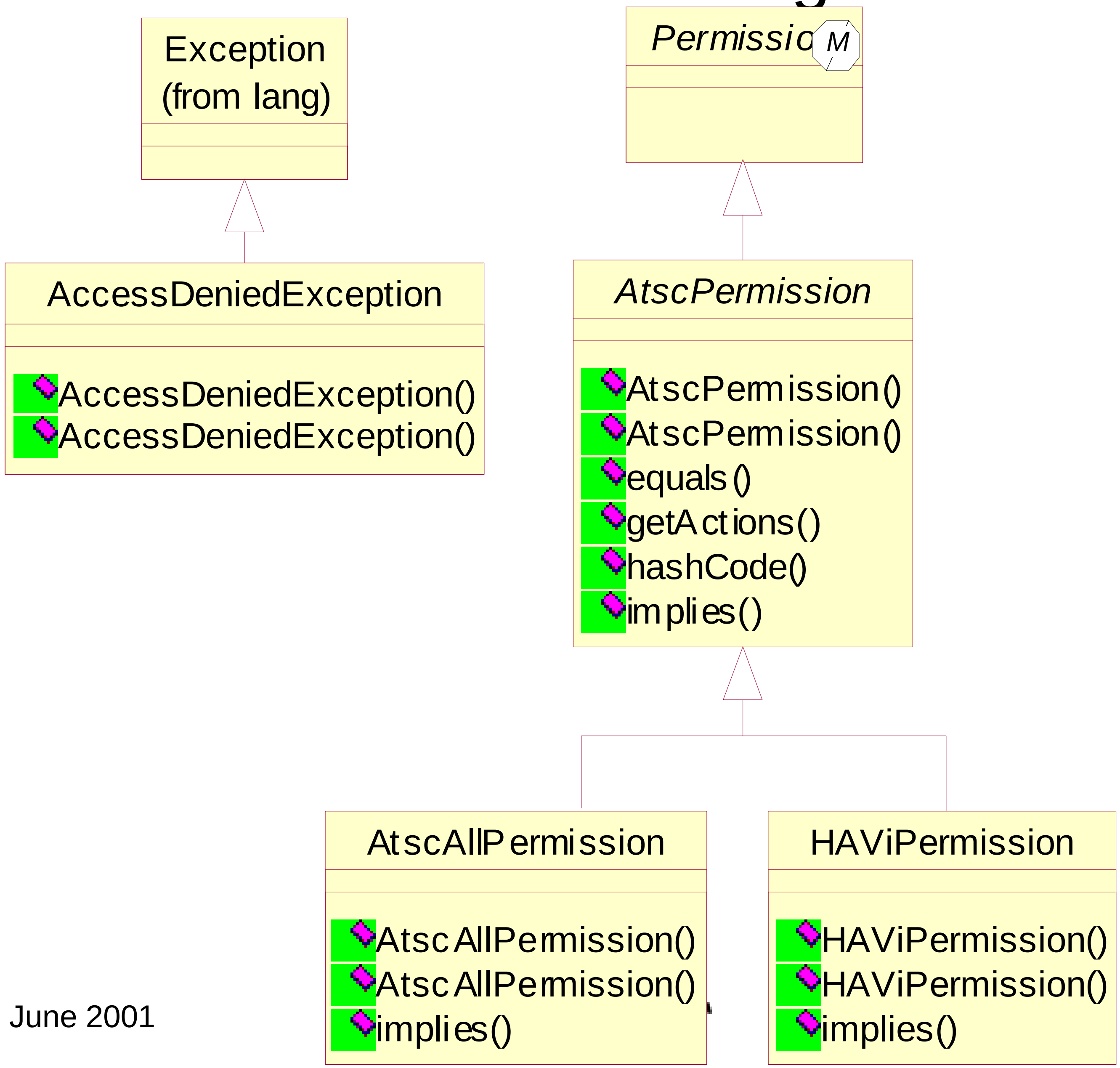
org.atsc.preferences



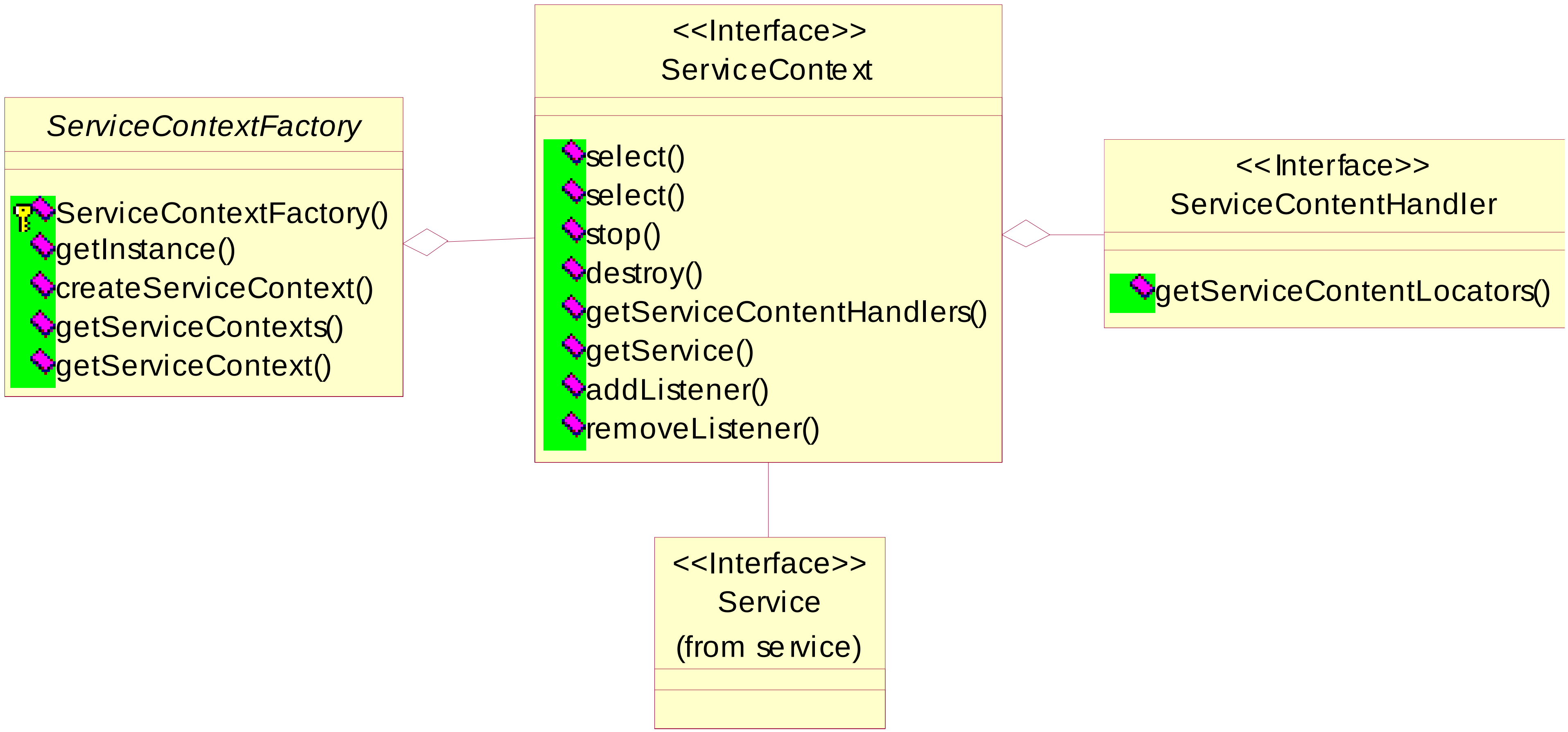
User Preference Scenario



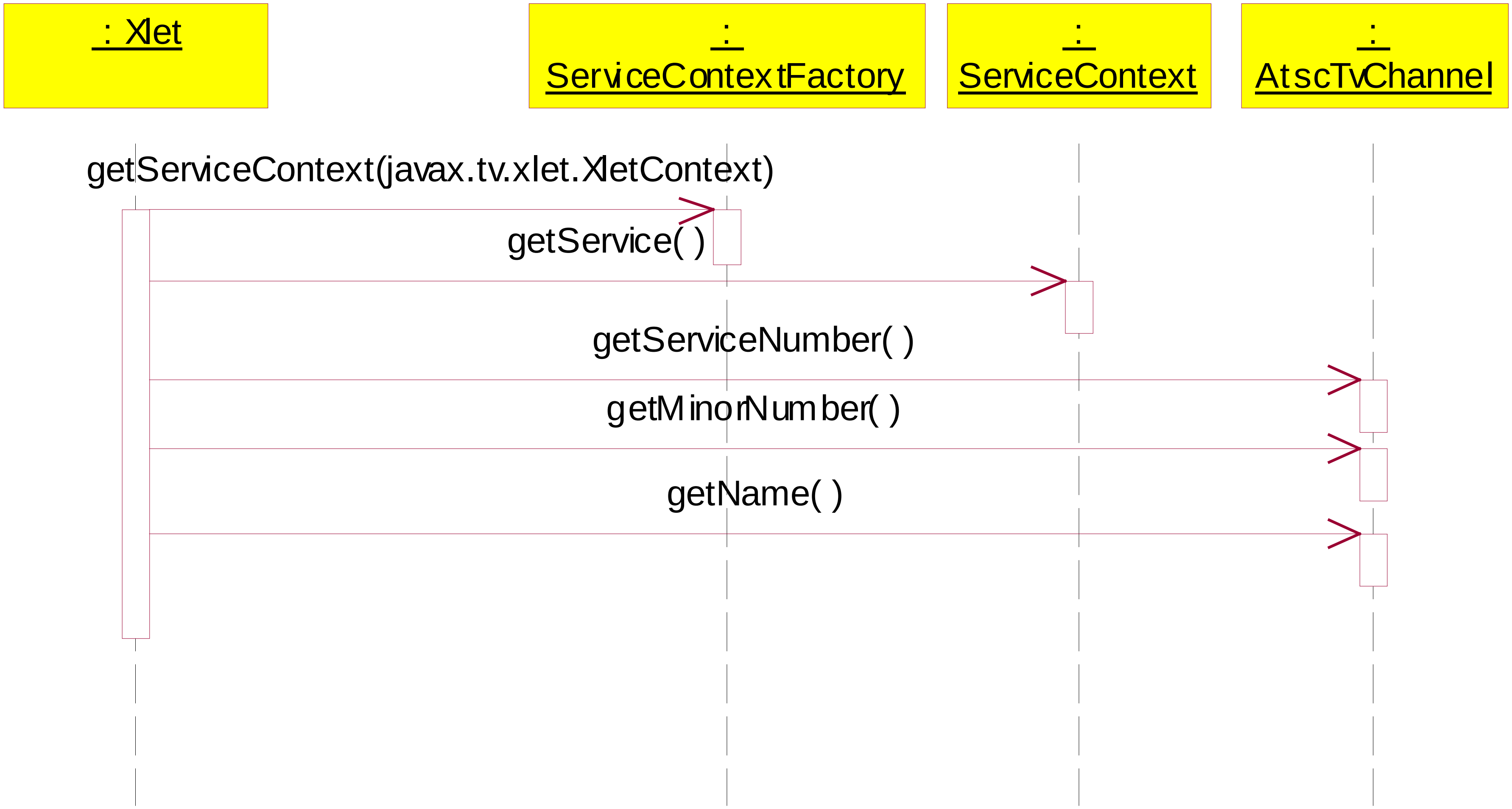
org.atsc.security



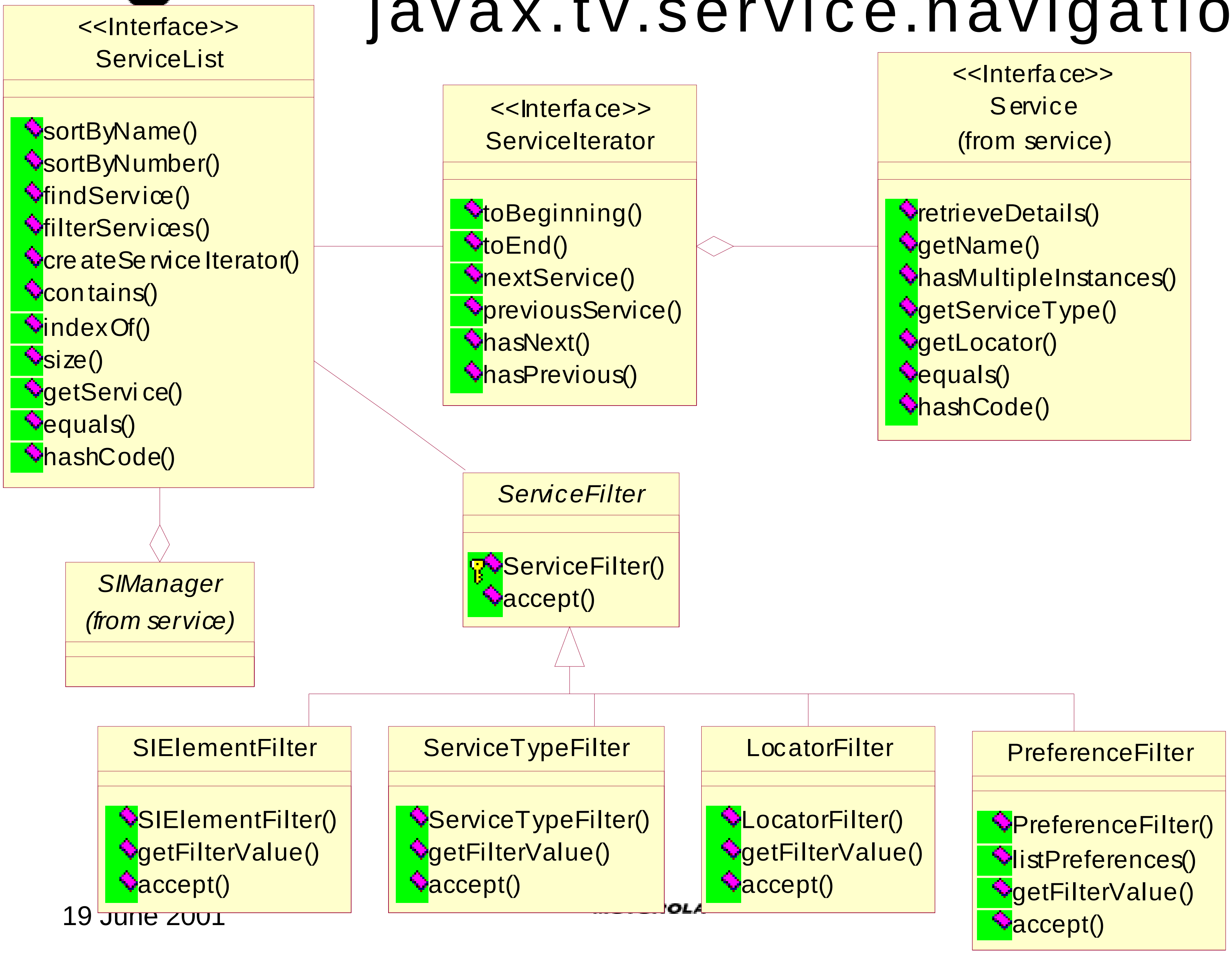
javax.tv.service.selection



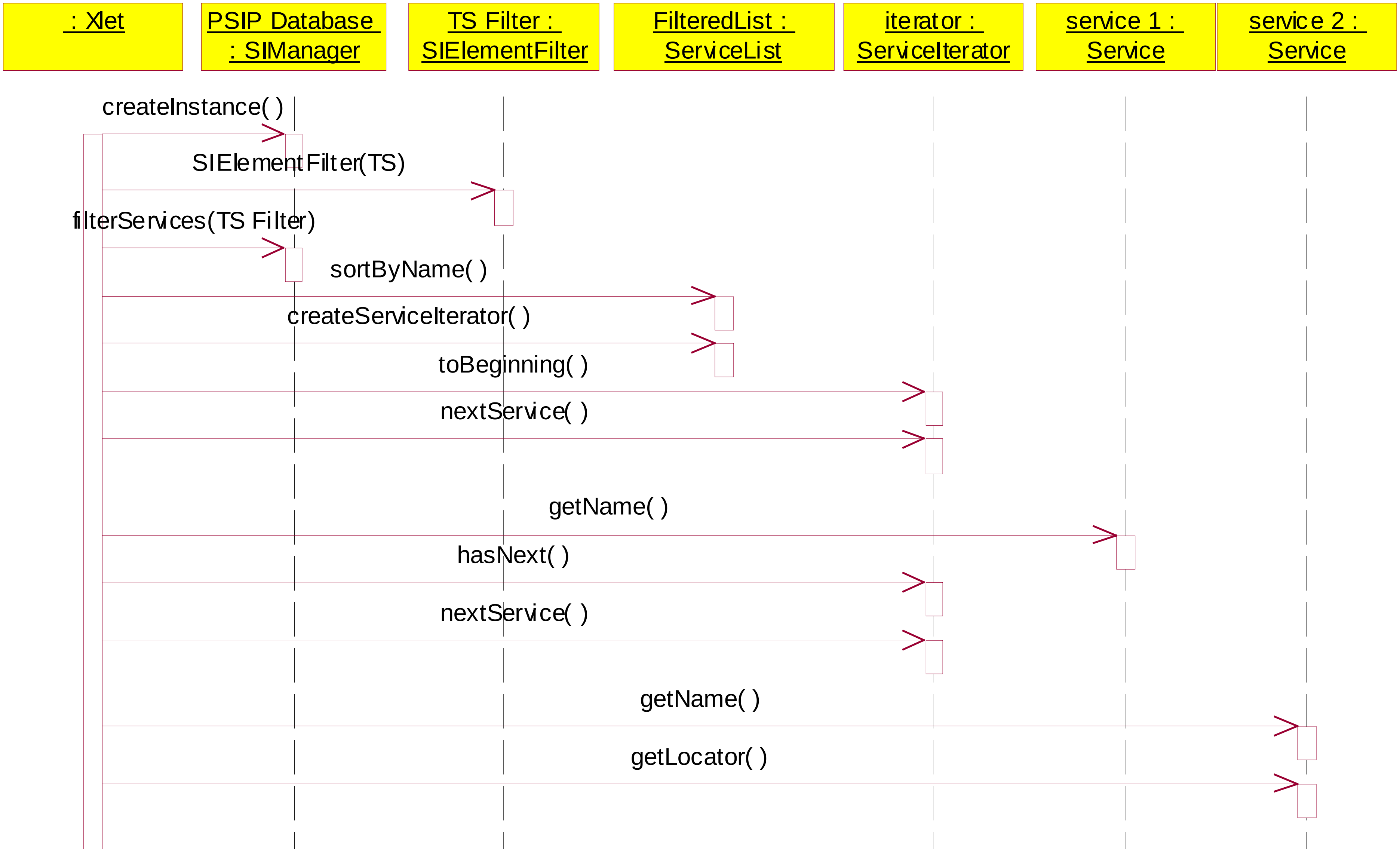
Discovering Current Service Scenario



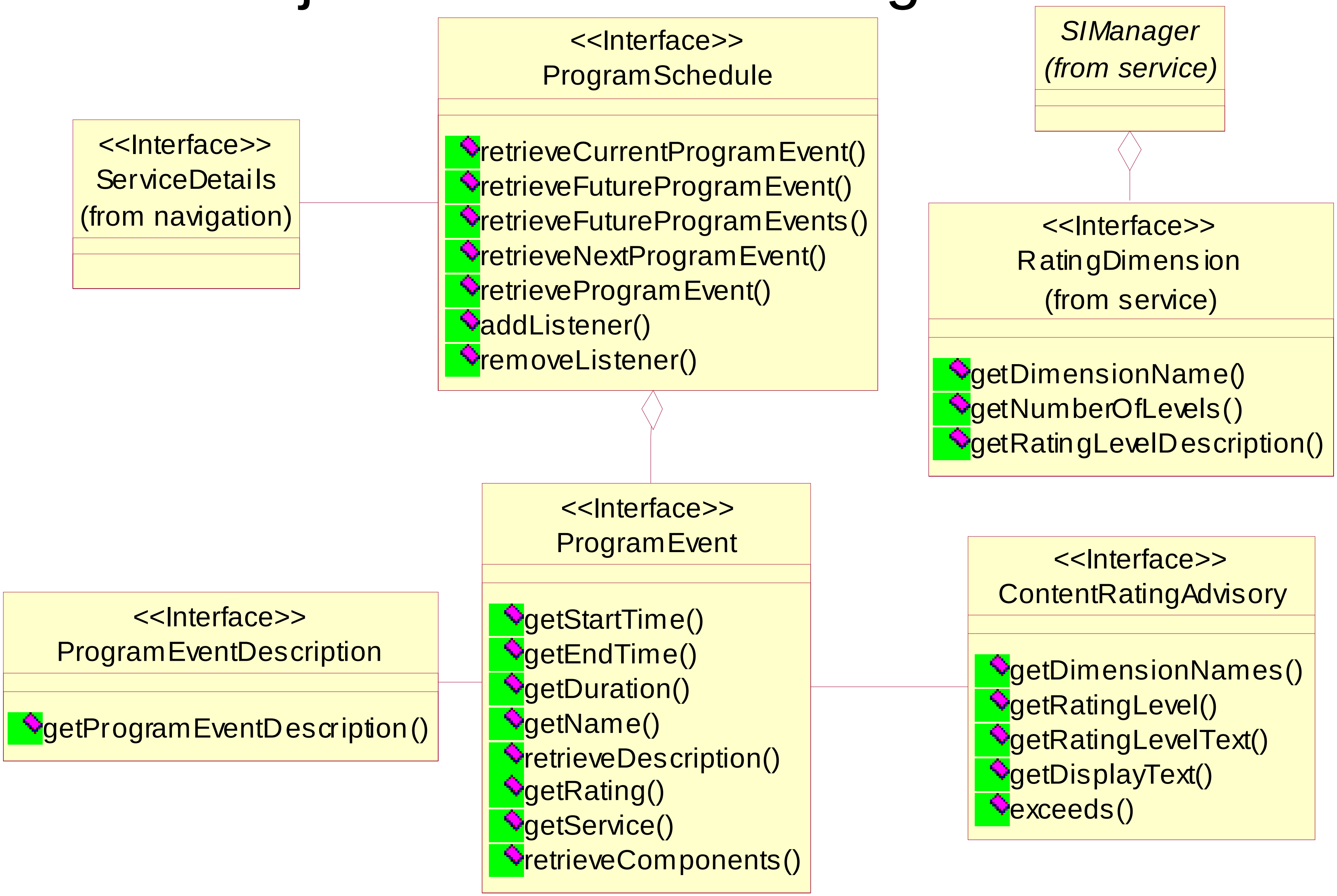
javax.tv.service.navigation



Service Collection Scenario



javax.tv.service.guide



Retrieving Current Program Information

Conclusion

- DASE provides a very rich set of Java APIs as part of the Procedural Application Environment
- The selected APIs are made to work together
- The Declarative and Procedural Applications may work together via the DOM APIs
- Content providers have a powerful environment to create compelling content

Java TV 1.0 API Technical Overview

Jim VanLoo

Sun Microsystems
James. Vanloo@sun.com

The objective of the javax.tv.* packages is to browse, select, and control broadcast content, both executable byte code and media streams. The package design requires certain broadcast protocol support, but abstracts the protocol so as to provide a single formalism for broadcast content and a single collection of interfaces to interact with such content. The scope of the technical overview is:

- Execution Environment:
  - Java Virtual Machine
  - Broadcast Independent (Implicit) Packages (java.*)
  - Broadcast Specific Packages (javax.tv.* and javax.media.*)
  - Silent on User Interface Packages (java.awt.* and org.havi.ui.*)
- Service Life Cycle:
  - Executable Content (javax.tv.xlet.*)
  - Media Content (javax.tv.service.*)
- Service Metadata
  - Service Portals (javax.tv. {navigation,guide,transport})
- Service Selection
  - Service
  - ServiceContext
  - ServiceComponents (with companion ServiceContentHandler)
- Data Selection
  - Broadcast Protocol Data (javax.tv.carousel.*)
  - Internet Protocol Data (java.net.* and javax.tv.net.*)
- Media Stream Control (javax.media.* and javax.media.protocol.*)

The Java TV 1.0 API: Technical Overview

Jon Courtney
Java TV Specification Lead
Sun Microsystems

Purpose of This Presentation

Become familiar with the primary features that the Java TV API provides for creating content for interactive digital television.

About the Speaker

- Led the completion of the Java TV 1.0 API specification.
- Represented Sun and promoted Java TV API in television standards bodies in U.S. & Europe.
- Currently specification lead for J2ME Personal Profile.

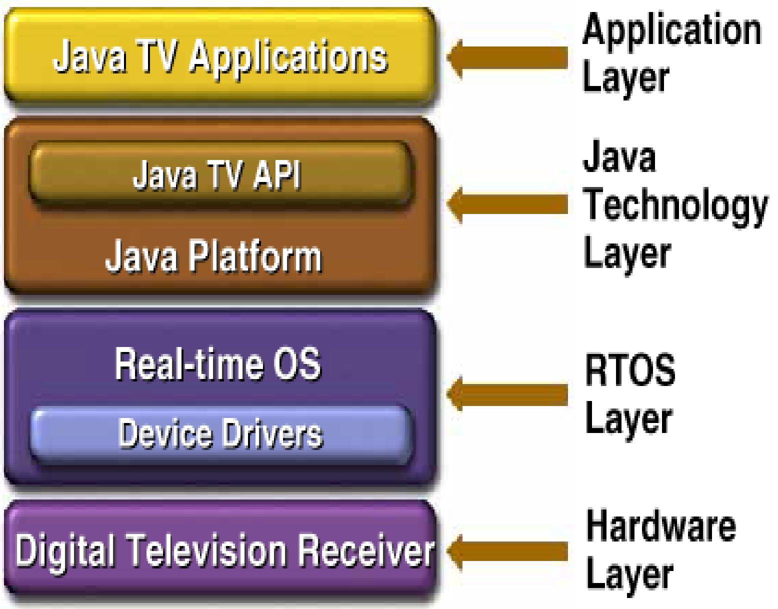
JavaOne

Key Topics

- Java TV Broadcast Environment
- Application Life Cycle Model
- Service Information API
- Service Selection API
- Broadcast Data APIs
- Media Control

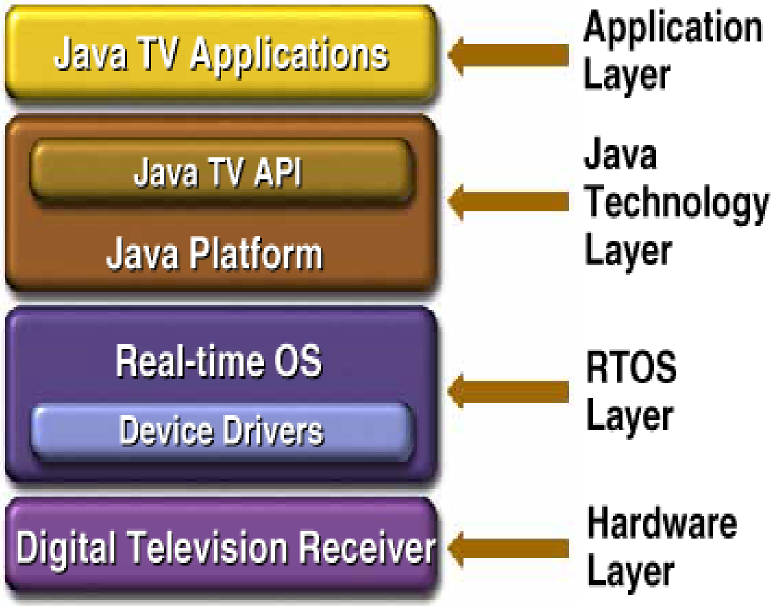
Environment

- Java Platform
    - Virtual Machine
    - Core APIs
    - UI APIs
    - TV extension APIs



Environment

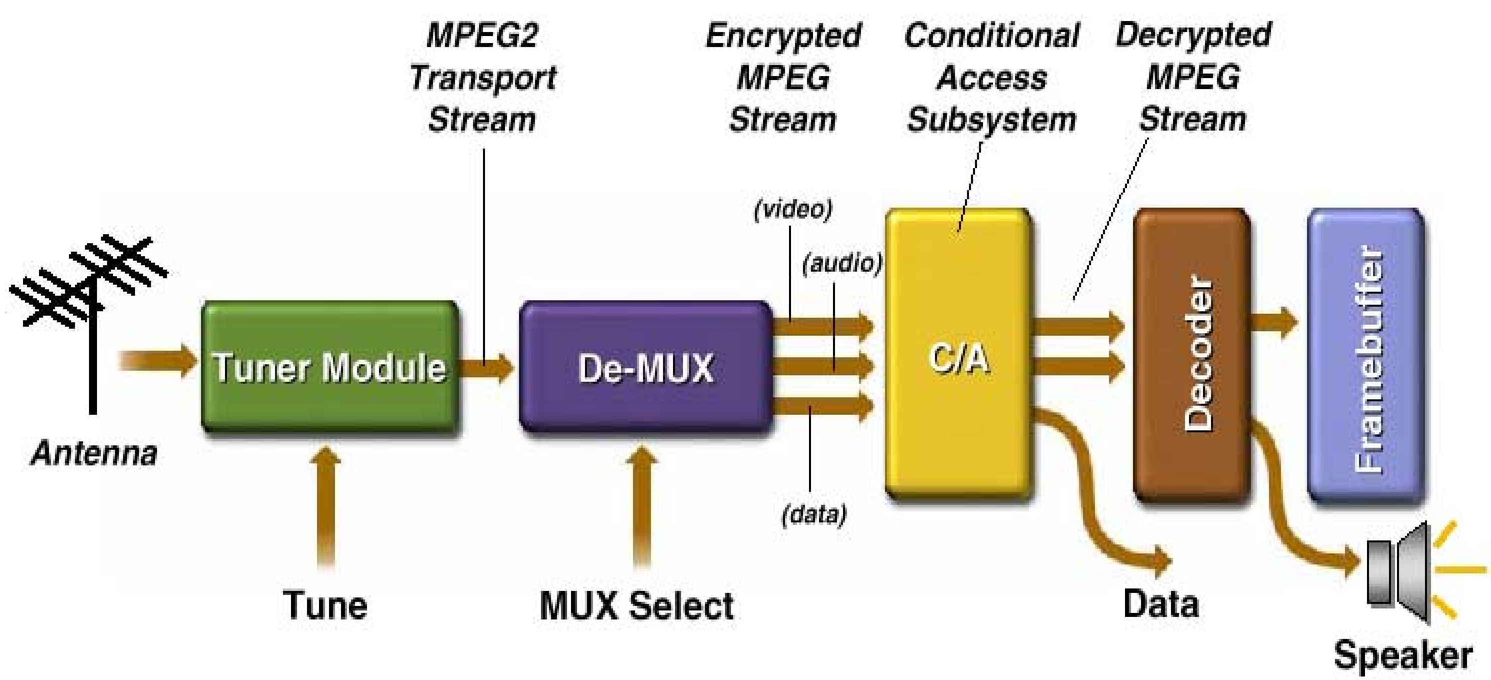
- Broadcast Platform
  - Operating System
  - Tuner Control
  - Demux Control
  - Conditional Access
    - Media Pipeline
    - Service Information Database



JavaOne

Broadcast Platform

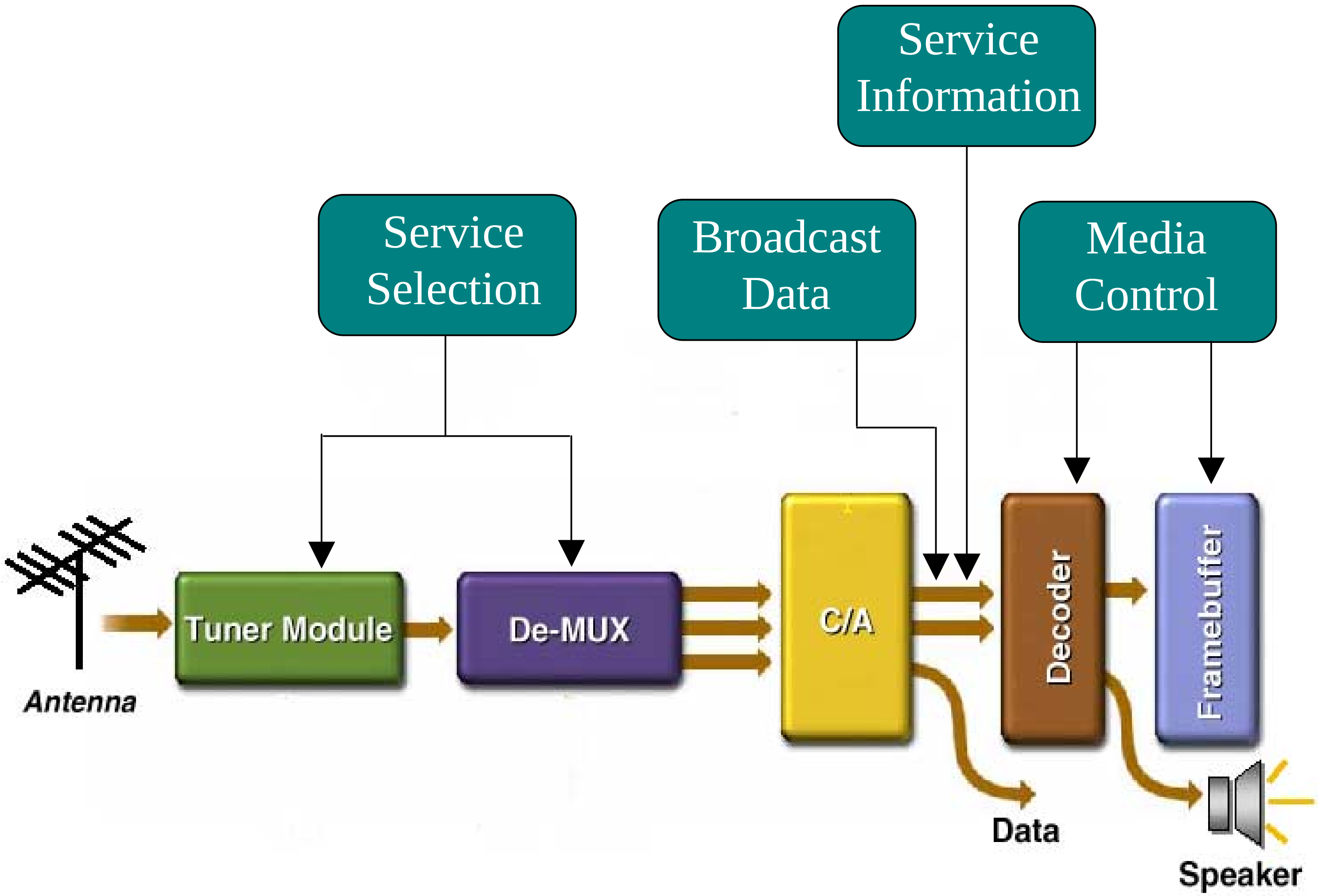
Major Hardware Components



Java TV

Architecture & APIs

Java TV Components



Java Platform Features

Basic services for TV applications

- Input/Output
  Java.io
- Networking
  java.net
- Graphics & UI
  Java.awt
- System functions
  java.lang, java.security, java.util...

Java TV Architecture

Major API Elements

- Application life cycle
- Service Information
- Service Selection
- Broadcast Data
- Media Control

Java TV Architecture

Locators

- A mechanism for referencing data and resources
- Locators are opaque references to
  - Broadcast file systems
  - Portions of service information
  - Sources of audio and video content
  - etc.

Java TV Architecture

Locators

- Handles to information and resources
- Typically generated by the API
- Created from / externalized to string form:
    - LocatorFactory.create(String) -> Locator
    - Locator.toExternalForm() -> String

Java TV Architecture

Security & Resource Management

- Policy is determined by network/platform
- Policy enforced by receiver
- Expressed using exceptions
- Try & refuse model

Java TV Architecture

Application Life Cycle Model

Application Life Cycle

Goal: Define a model for TV applications

- Learn from existing application models
- Develop a model appropriate for TV

Application Life Cycle Model

Features:

- Ease of use for application developers
- Model separate from:
  - Window system management
  - Resource management
  - Application management policy
- Minimal requirements on app managers

Application Life Cycle Components

- Application Manager
- Xlet
- XletContext
- *Xlet State Machine*

Application Manager

- Xlets can be destroyed at any time
- Current state of Xlet will always be known
- An Application Manager can change an Xlet's state
- An Application Manager will know if an Xlet has changed state

Application Life Cycle

Four application states:

- Loaded
  - Code is loaded, initialized
- Paused
  - Application quiescent, minimal resource usage
- Active
  - Application is executing normally
- Destroyed
  - Application has released resources, terminated

Application Life Cycle

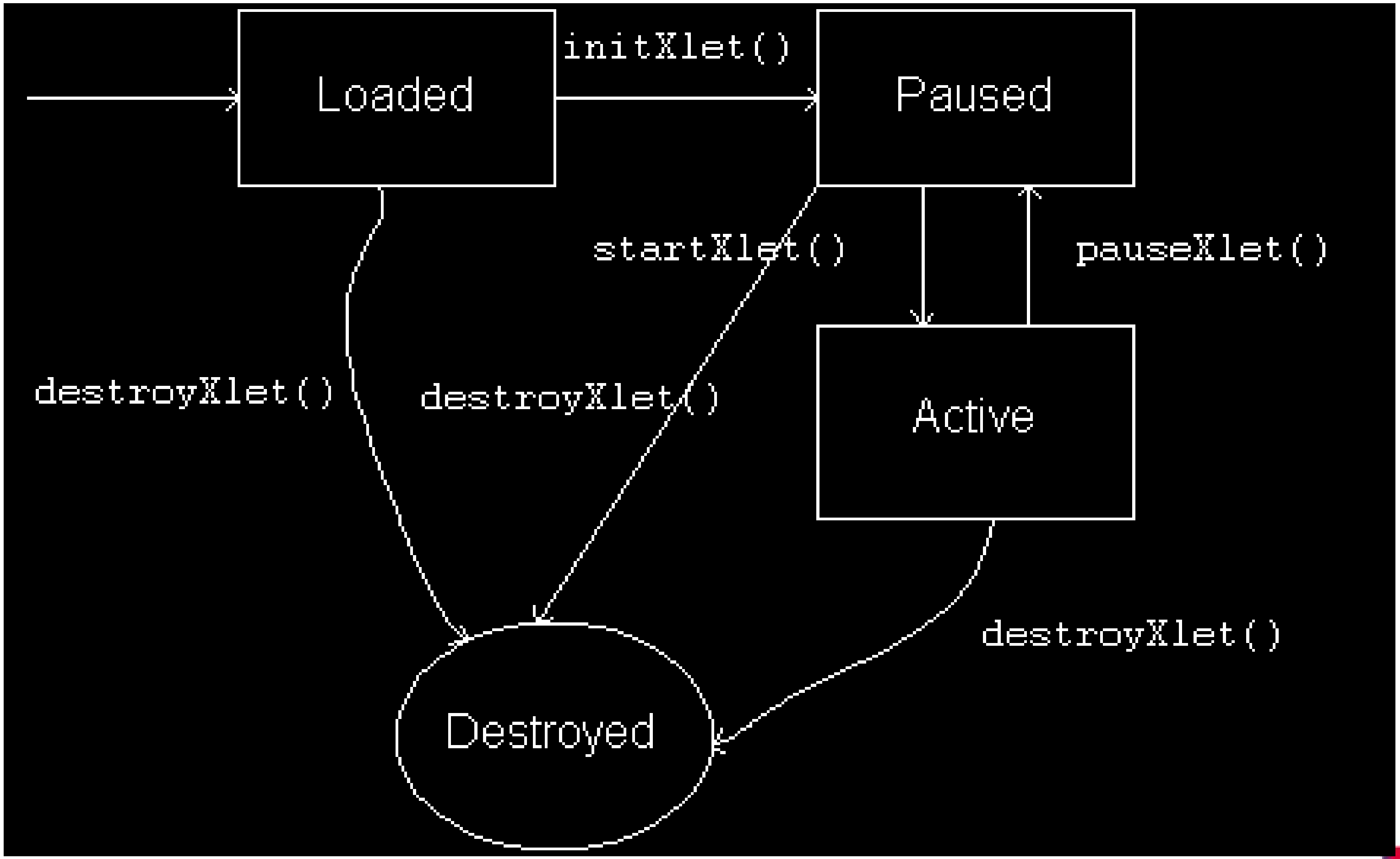
Xlet Interface

- Implemented by the application
- Methods to signal state transitions
- Xlets managed by Application Manager
- Similar to applet model w/o UI

Application Life Cycle

```
Package javax.tv.xlet;
public interface Xlet {
  void initXlet(...);
  void pauseXlet();
  void startXlet();
  void destroyXlet(...);
}
```

Application Life Cycle



Application Life Cycle

XletContext

- Provides property interface
- Used by Xlet to signal state transitions to the application manager
- Xlet.initXlet(XletContext context);

Application Life Cycle

```
package javax.tv.xlet;
public interface XletContext {
  Object getXletProperty(String);
  void notifyPaused();
  void resumeRequest();
  void notifyDestroyed();
}
```

Java TV Architecture

Service Information API

Service Information

- What is Service *Information*?
  - Data in the broadcast stream
  - Provides details about the available services
- What is a *Service*?
  - A collection of content for display
  - Audio/Video/Applications/Data
  - Often referred to as a "channel"

Service Information

- Data format is protocol independent
- Accessible to applications via SI API
- SI model is read-only database
- Database populated from the broadcast

JavaOne™

Service Information API

Features

- Protocol independent
- Storage and delivery independent
- Extensible for new SI types
- Cached and non-cached access
- Sync and async access
- Service discovery

Service Information API

Three "views" of service information:

- Navigation package
  - Traversing through hierarchical SI data
- Guide package
  - EPG support
    - Program schedules, events, rating info
- Transport package
  - Exposes SI delivery mechanisms

Service Information API

Asynchronous Retrieval

- Database cannot cache all SI data
- High latency in accessing data not in cache
- Inconvenient for programs to block while waiting for data

Service Information API

Asynchronous Retrieval

- Asynchronous retrieval mechanism permits applications to queue requests and continue execution
- Asynchronous data access methods prefixed with 'retrieve':

RetrieveProgramEvent(...)

Service Information API

Asynchronous Retrieval

" Interface SIRequestor implemented by applications to receive data

void notifySuccess(SIRetreivable[])

void notifyFailure(...)

Service Information API

Asynchronous Retrieval

- Interface SIRetrievable extended by retrievable data types
  - Bouquet
  - Network
  - ProgramEvent
  - ServiceDetails
  - Etc.

Service Information API

- SIRequest objects returned by asynchronous retrieval calls

  Boolean cancel();

- Example:

  SIRequest retrieveProgramEvent(Locator, SIRequestor);

Service Information API

Request model - summary

- Objects wishing to receive service information asynchronously implement SIRequestor
- Data is returned as SIRetrievable
- SIRequest objects returned to cancel the request

JavaOne

Service Information API

SI Manager

- Provides access to SI database
- Event generator describing SI updates
- Provides list of available services
- SI filtering operations

Service Information API

```
Package javax.tv.service.navigation;

public class SIManager {

  ServiceCollection
    createServiceCollection(ServiceFilter);

  Service getService(Locator);

  Transport[] getTransports();

  SIRequest retrieveSIElement(Locator, SIRequestor);

}
```

Service Information API

Service API

- Represents a source of content, aka "channel"
- Selectable via *service selection* API
- Persistent data: name/number, locator
  - Cached available synchronously
  - "Installed services" for bootstrap
- Asynchronous access to service "details"

Service Information API

ServiceDetails

- Service meta-data
  - Represents a specific instance of a service in the broadcast
  - Reports description, program schedule, etc.
  - Reports service components & types (e.g. Audio, video, data)
- Extensible for new meta-data

Java TV Architecture

Service Selection API

Service Selection

Features

- Abstracts "tuning" operation
- Asynchronous operation
- Conditional access results exposed
- Support for multiple selection "contexts"

JavaOne

Service Selection

Key APIs

- ServiceContext
  - Object used to *select* a service
  - Often maps to a physical tuner on the device
- ServiceContentHandler
  - Responsible for the presentation of a service
  - Typically related to a JMF Player

Service Selection

ServiceContext

- Represents an environment for presenting media and downloaded applications in a service.
- Provides service selection operation

  ServiceContext.select(Service);
- Reports currently selected service

Service Selection

ServiceContext

- Management of multiple contexts
- Access to content "handlers"
- Signals current state via events for completion, redirection, failure

Service Selection

Service Context State Model

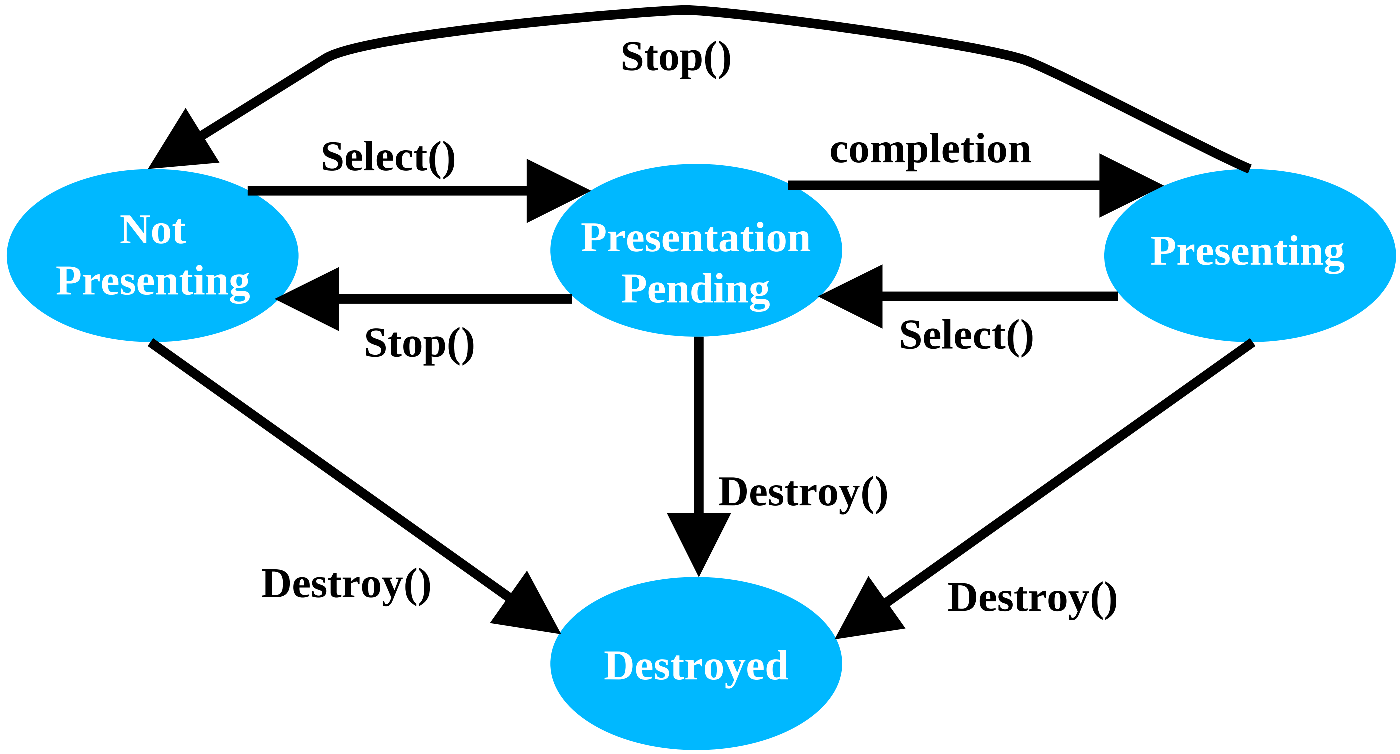
- Not Presenting
  - PresentationTerminatedEvent
- Presentation Pending
  - After select operation, before completion

Service Selection

ServiceContext State Model

- Presenting
  - NormalContentEvent: Requested content is presented
  - AlternativeContentEvent: C/A redirection
- Destroyed
  - ServiceContextDestroyedEvent

Service Context States



Java TV Architecture

Broadcast Data APIs

Broadcast Data

Features

- File style access to broadcast filesystems
- Push style delivery for streams
- DatagramSocket access to broadcast IP

Broadcast Data

Package javax.tv.carousel

- Provides access to bounded data in hierarchical, cyclically transmitted broadcast filesystem
  - DSMCC object carousel
  - DSMCC data carousel
  - ATVEF UHTTP

Broadcast Data

Package javax.tv.carousel

- CarouselFile extends java.io.File
    - Represents broadcast files
    - Familiar mechanisms from java.io package
        - FileInputStream
        - RandomAccessFile
        - FileReader

Broadcast Data

CarouselFile

- Event notification of content changes
  - Interface CarouselFileListener
- Latency management
  - Instancing a CarouselFile notifies system to asynchronously cache file from broadcast
- Referenced via locators or filenames
  - Broadcast filesystem is mapped into local file name space

Broadcast Data

PushSourceStream

- Represents source of streaming data
- Acquired through JMF manager
- Delivers data in non-flow-controlled manner
  - Client is notified when data arrives
- Subinterface throws exceptions for data loss

Broadcast Data

Package javax.tv.net

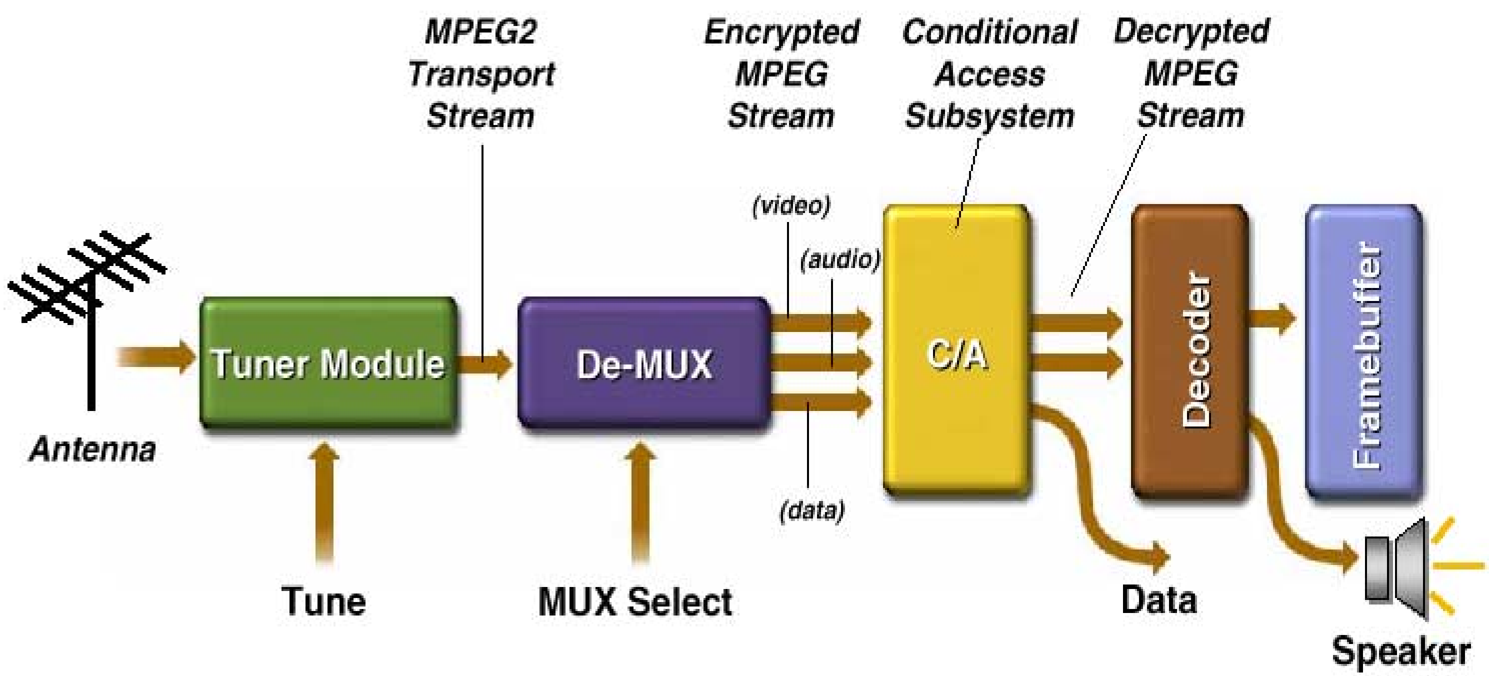
- javax.tv.net.InterfaceMap permits access to broadcast IP through conventional mechanisms
  - Dynamically maps locator to broadcast IP into private local IP address
  - Unicast and multicast supported
  - Access through familiar java.net mechanisms
    - DatagramSocket, MulticastSocket

Java TV Architecture

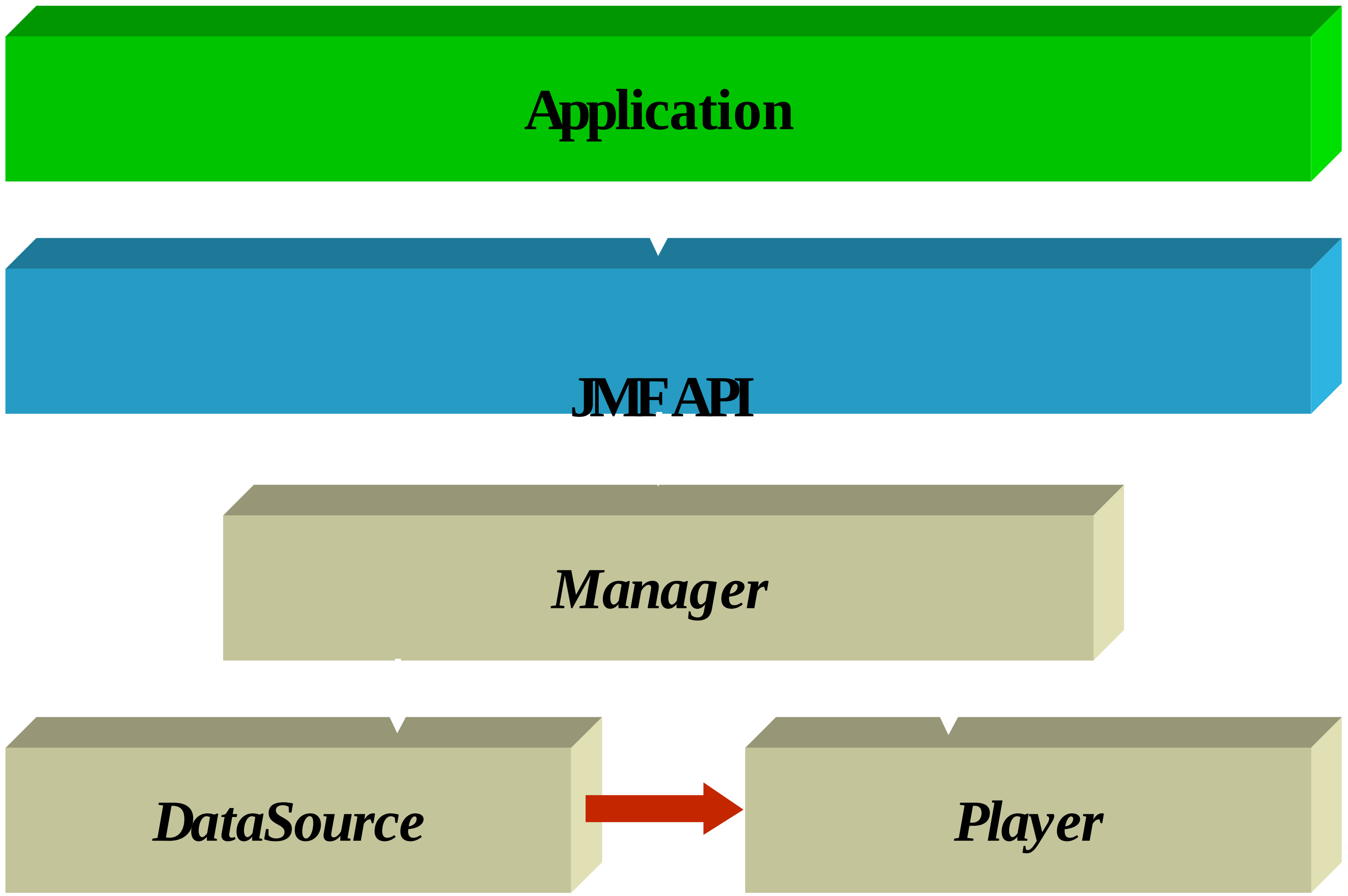
Media Control APIs

Media Control

- Java Media Framework manages pipeline
- JMF Player wraps decoder, rendering
- JMF DataSource wraps tuner & demux



JMF Architecture



Media Control

- Player
  - Renderer of streaming content
  - Supports one or more media types
  - Likely implemented in hardware
  - Manages state and synchronization

Media Control

- Controller
  - Subinterface of Player
  - Provides state change notification
  - Manages state machine

Media Control

- DataSource
  - Abstracts the source of the media data
  - Data is typed
  - Location of data referred to by an opaque reference

Broadcast Pipeline

JMF Player and DataSource

- Representation of network interface
- Representation of rendering pipeline
- Separation allows reuse of pipeline
- Synchronization primitives
    - Media time exposed

Broadcast Pipeline

JMF Player and DataSource

- A/V control primitives
  - JMF Controls published
  - Runtime extendible
  - Media time control
- Resource management mechanisms
  - Events signal state transitions
- Small framework abstracts hardware

JMF and Java TV

- JMF mostly hidden to applications
- DataSource & Player connected transparently
- When a service is *presenting,* the JMF Players can be obtained

    SC.getServiceContentHandlers();

- Some standards define their own JMF controls

Java TV Architecture

Additional APIs

Graphics APIs

- AlphaColor
  - Subclasses java.awt.Color
  - Provides a simple alpha blending color
- TVContainer
  - Provides Xlets with a root graphics container

Timer API

- Provides support for timed events
- Allows applications to be called after a particular time has elapsed
- Similar to PersonalJava pTimer API

Additional Information

- Java TV product web page

  java.sun.com/products/javatv

JavaOne[SM]

Sun's 2001 Worldwide Java Developer Conference

Q&A

JavaOne℠

Sun's 2001 Worldwide Java Developer Conference™

ATSC Digital Television MPEG, PSIP and Data Broadcast

Rich Chernock

IBM Research Watson
Research Center Hawthorne,
NY
chernock@raleigh.ibm.com

This presentation will give a broad overview of the ATSC system layers that DASE builds upon and assumes to be present. While not an in-depth examination of these layers, the intent is to give the audience an idea of the framework that they can count on and must use. The topics to be covered are: MPEG-2 Systems, PSIP and Data Broadcast.

MPEG-2 Systems: The base "plumbing" layer that ATSC utilizes for the transport of all broadcast data. MPEG-2 systems provides the multiplexing and encapsulation structures to carry data for DASE applications, as well as "metadata" (PSI) necessary to unwrap the different broadcast components.

PSIP: Program and System Information Protocol, which is used in ATSC systems to allow receivers to discover what components are in the broadcast, link to the resources and provide program guide functionality to the viewer.

Data Broadcast: The T3-S 13 data broadcast standard (Al90) which specifies how to encapsulate data for broadcast on an A TSC system, as well as the mechanisms for announcement (figure out what will be broadcast) and signaling (locate and bind the resources for a data service).

ATSC Digital Television: MPEG, PSIP and Data Broadcast

Rich Chernock

IBM Research

chernock@raleigh.ibm.com

With thanks to:
Pete Schirling, IBM
Art Allison, NAB
Regis Crinon, Intel
Michael Isnardi, Sarnoff

Environment

The Old Days: Television involves wiggling voltages in the right way at the right times so the receiver can recreate the pictures

The New Paradigm: Television involves transmitting database information and parameters to allow the pictures to be calculated.

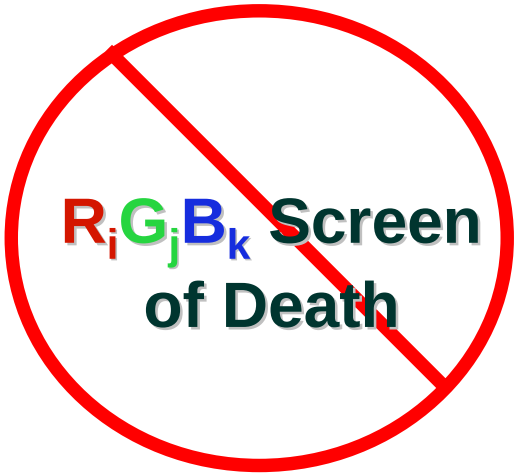
Observations:

This is NOT a computer network

A TV is not expected to behave like a computer

Going black is NOT an option

Assets not available locally or in the broadcast don't exist



MPEG-2 Systems Overview

Pete Schirling

Senior Consulting Engineer

IBM Research

Digital Media Standards and Commercialization

River Road MS 863N
Essex Jct, VT 05452
Phone -+1 802 769 6123
Fax -+1 802 769 7362
e-mail -schirlin@us.ibm.com

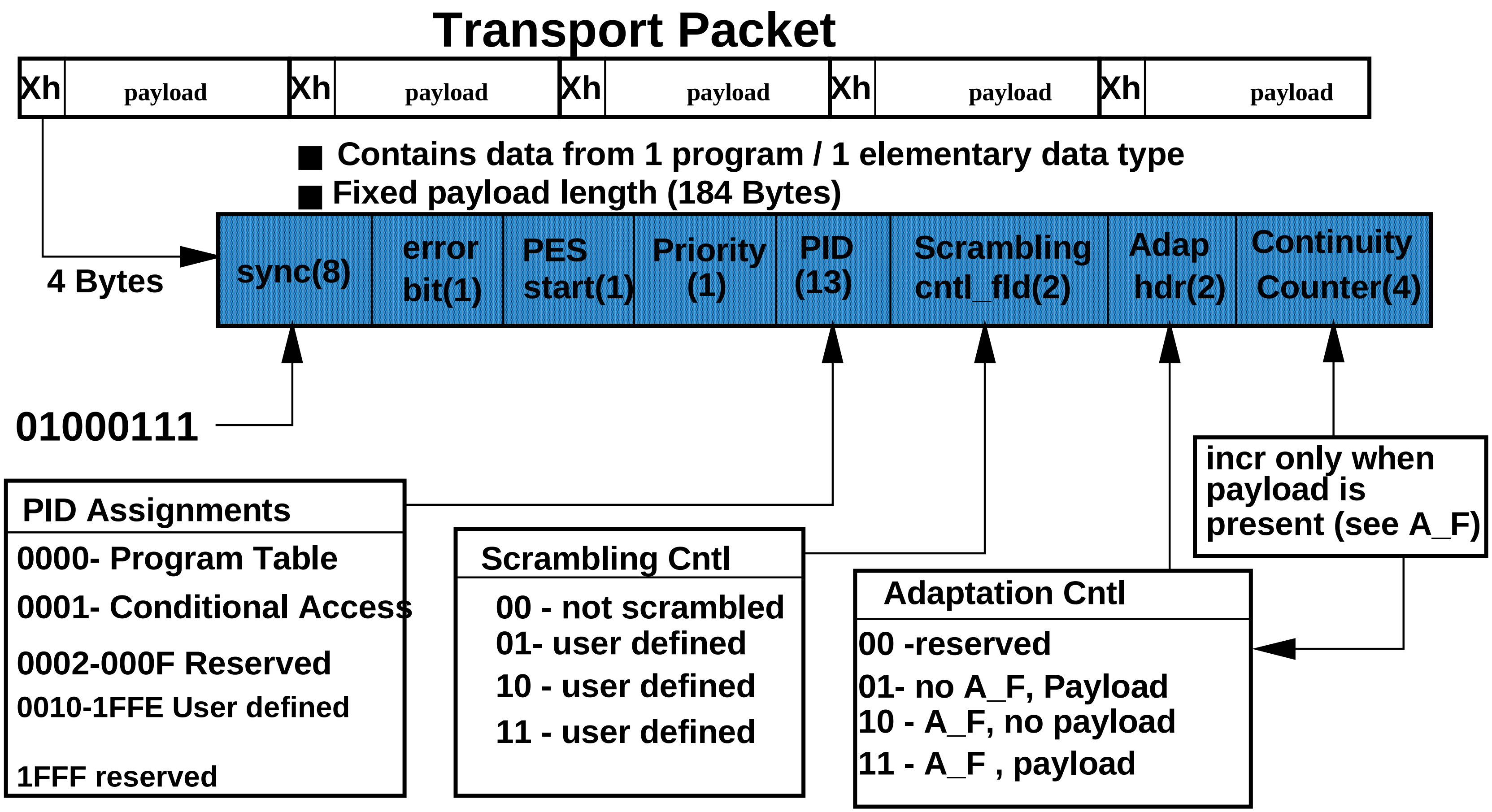
MPEG-2 Systems

Everything you didn't want to know but needed to in order to keep your job !!

MPEG-2 Transport Stream

- ***enabled as a universal carrier of real-time and non-real time information***
  - ✓ Multiple programs
  - ✓ Associated program information
    - PSI (Program specific information)
    - other information program or non-program related
  - ✓ Private or public information
    - Conditional access
    - Network or application specific

MPEG 2 Data Stream Definition



MPEG-2 Program Specific Information (PSI)

- Program Association Table
  - Links the MPEG-2 program_number with the PID carrying its TS_program_map_section
- Conditional Access Table
  - Carries CA_descriptors that point to the PID carrying the conditional access vendor's Entitlement Management Message (EMM) stream
- Program Map Table
  - Formed by the aggregation of all TS_program_map_sections contained in an MPEG-2 Transport Stream
  - Each MPEG-2 Program's TS_Program_map_section contains the "Program Definition"
  - The "Program Definition" specifies the Program Elements and descriptors associated with the MPEG-2 Program
- Transport Stream Description Table
  - Carries descriptors scoped to the entire transport stream

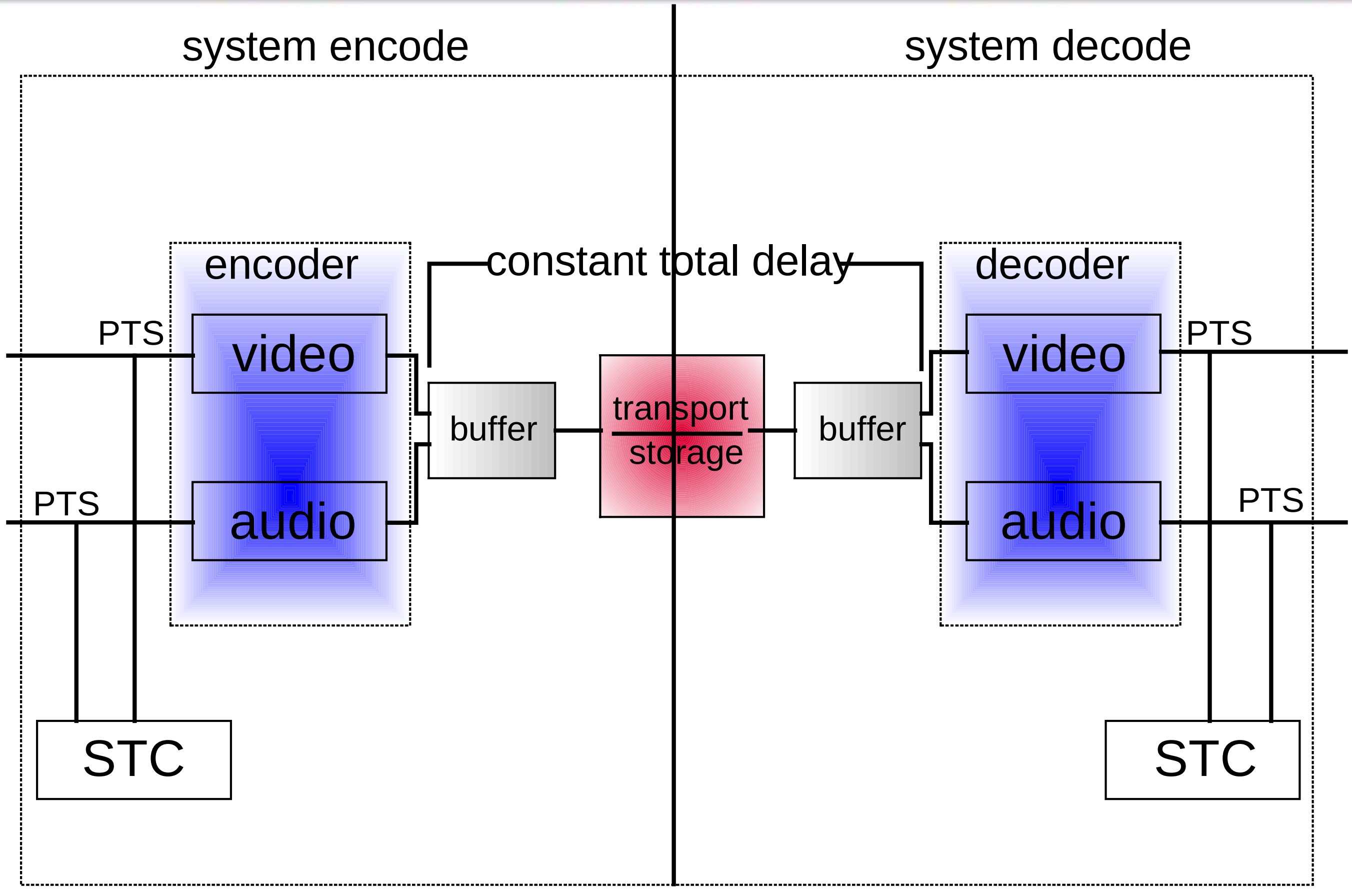
MPEG-2 as a clocked multiplex

- delivery is based on a constant delay model
- decoder system clock is carried in the stream
- decoder resource management is based on STC
- decoder synchronization is based on STC

CBR vs VBR

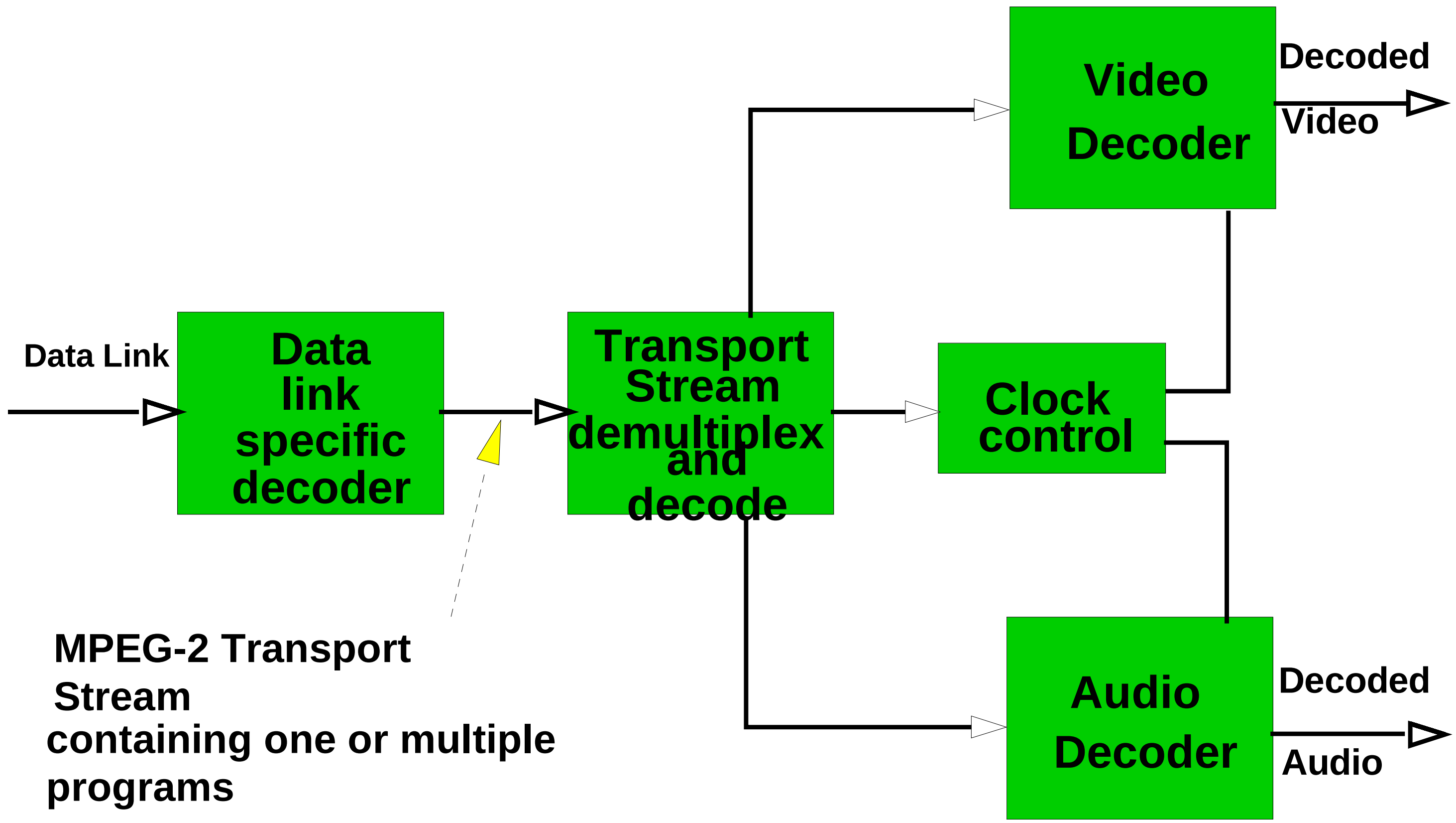
- CBR - Constant bit rate
  - non-variant byte stream
- VBR - Variable bit rate
  - piecewise constant bitrate
  - used in statistical multiplexing applications

Audio/Video Synchronization

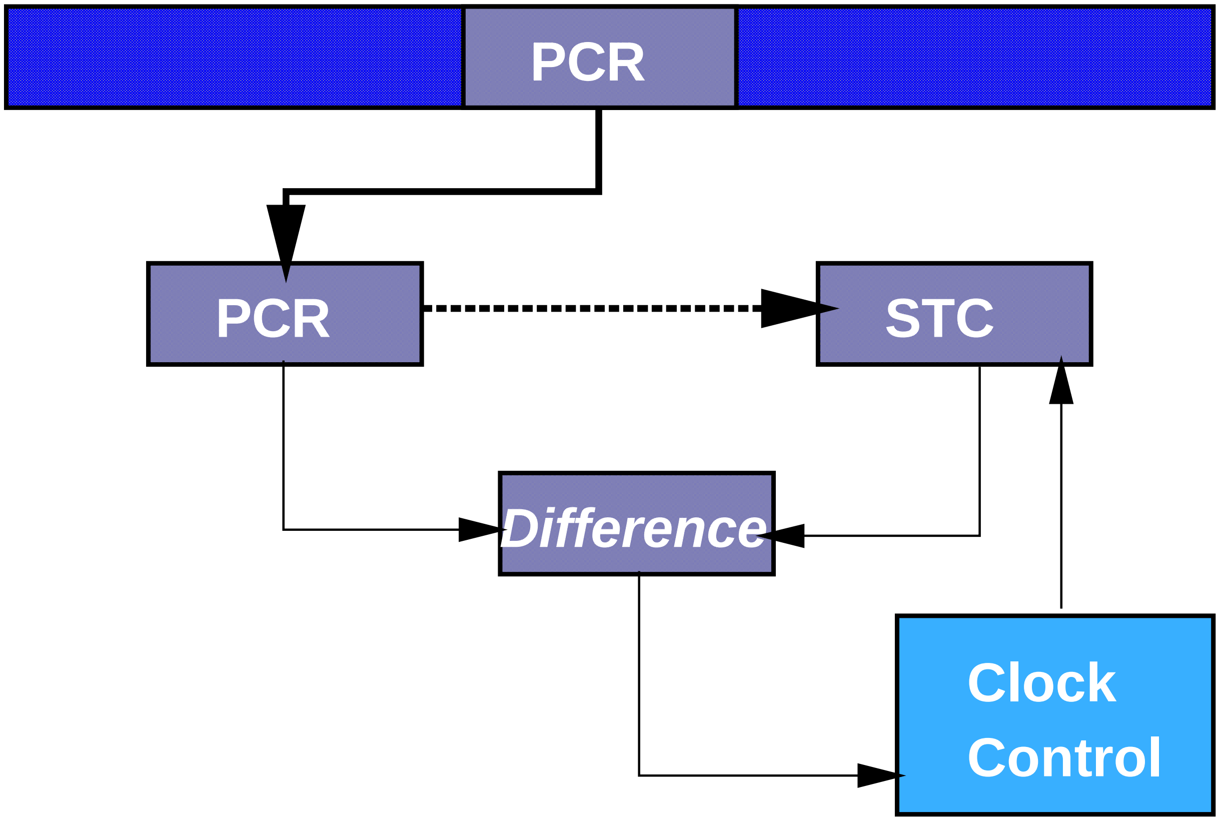


STC = System Time Clock

Transport Demultiplexing example



Clock Recovery



Buffer Management

- STC runs too fast -
  - caused by early byte arrivals
  - decoder runs too fast
  - causes buffer underflow
- STC runs too slow -
  - caused by delayed byte arrivals
  - decoder runs to slow
  - causes buffer overflow
- results in FRAME SLIPPING

Program and System Information Protocol for Terrestrial Broadcasting and Cable

P S I P

Art Allison

Director, Advanced Engineering

Science and Technology

National Association of Broadcasters

Why do we have PSIP?

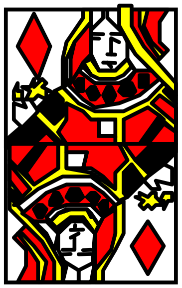
&

What is PSIP?

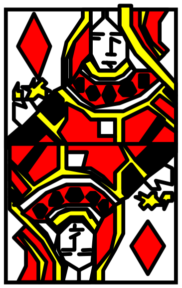
Legacy System

Pick your channel

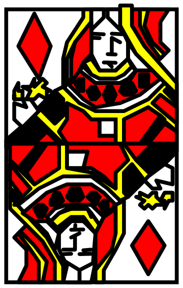
47



48

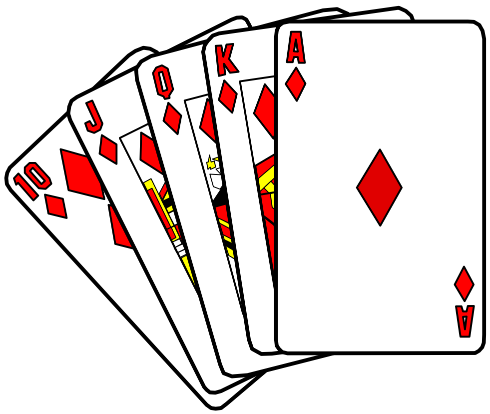


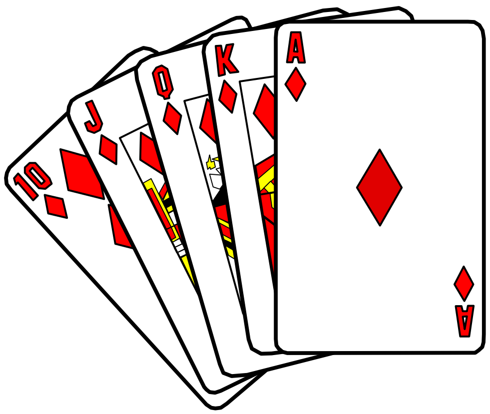
49

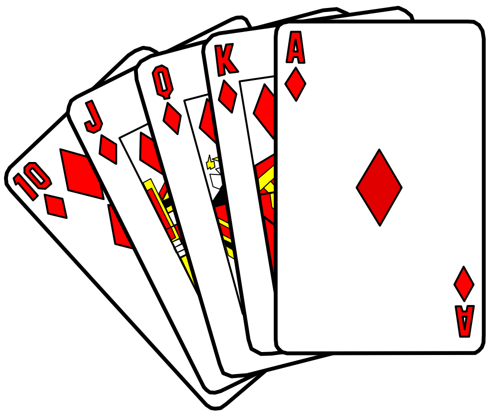


DTV System

Pick your Hand







47 48 49

Requirements

- Preserve channel number branding
- Support direct access to any channel
- Harmonized between terrestrial broadcast and cable TV
- Compatible with printed program guides

Requirements

- Support a variety of user-friendly navigation paradigms

Support grouping of digital and analog services

Extensible to data broadcasting and other services

PSIP Defined

- PSIP = Program and System Information Protocol
- Defined in ATSC Standard A/65A and Amendment 1 to A/65A
- Combines and Compacts A/55 and A/56
- Must be transmitted by ATSC terrestrial broadcasters in their DTV Transport Stream

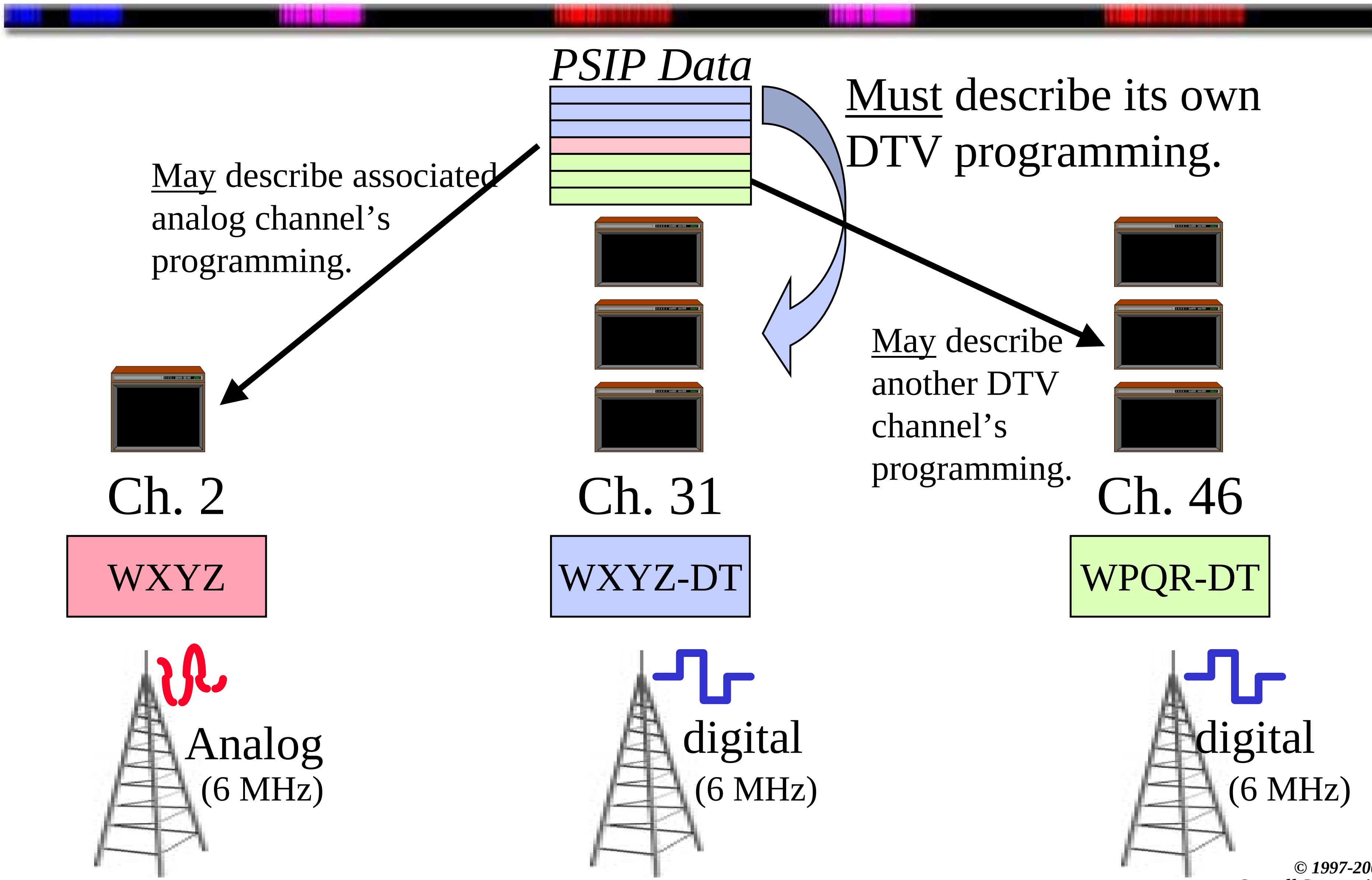
What PSIP provides

- Leverages existing broadcaster brand names
  - Maintain your channel identity
- Enables faster tuning
- Supports V-Chip and conditional access
- Also Provides an announcement service
  - Simple enough to go in every receiver
  - Extensible for higher end products
  - Small change in tuning paradigm for consumer
  - Compatible with printed media

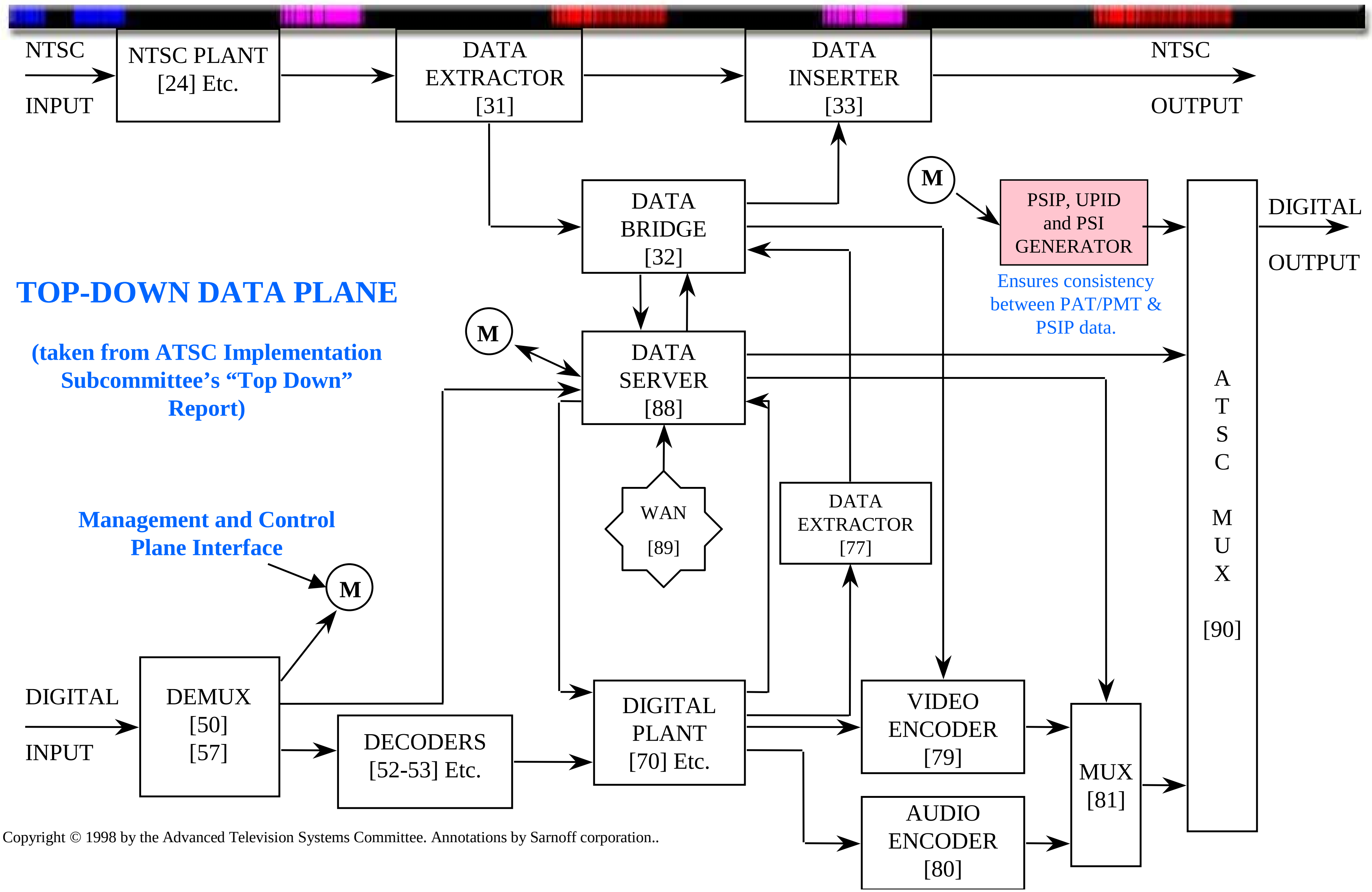
Complementary Functions Bind and Announce

- Main Table for current conditions
  - Contains the Transport Stream Linkages
  - Supports Tuning of the Programs Now
- One Main Table that describes LATER
  - Contains Future Event Announcements
    - (and in-progress events)
  - Enables Basic Electronic Program Guide
- Supporting Tables

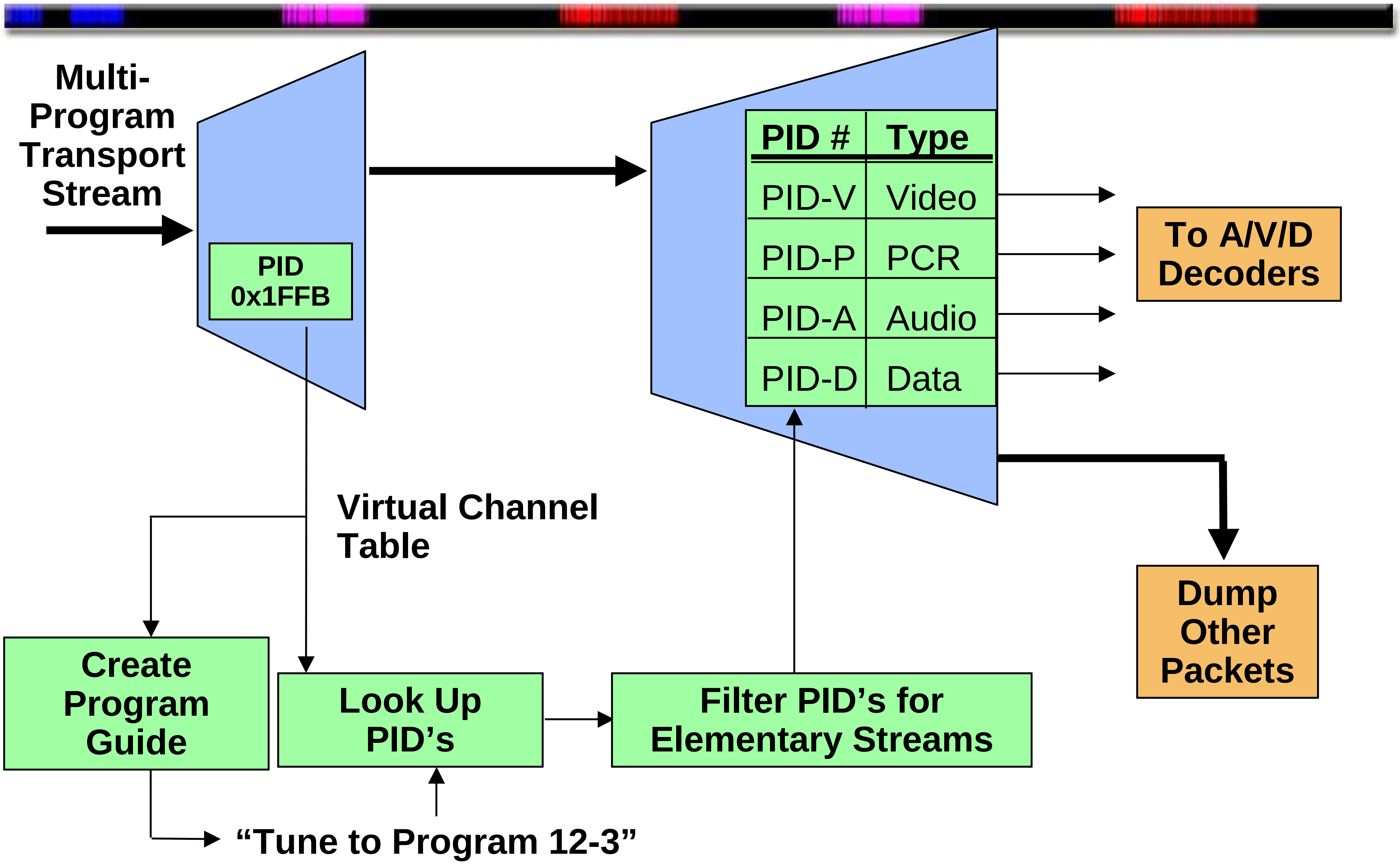
Scope of PSIP



PSIP Generation/Insertion



Tuning Example - PSIP



PSIP Tables

STT	System Time Table - provides time
MGT	Master Guide Table - provides version, size and PID's of all other tables (except STT)
VCT	Virtual Channel Table - provides attributes for all virtual channels in this Transport Stream
RRT	Rating Region Table - provides rating information for multiple geographic regions
EIT	Event Information Table - provides information for events on the virtual channels
ETT	Extended Text Table - provides detailed descriptions of virtual channels and events
DCCT	Directed Channel Change Table
DCCST	Directed Channel Change Selection Code Table

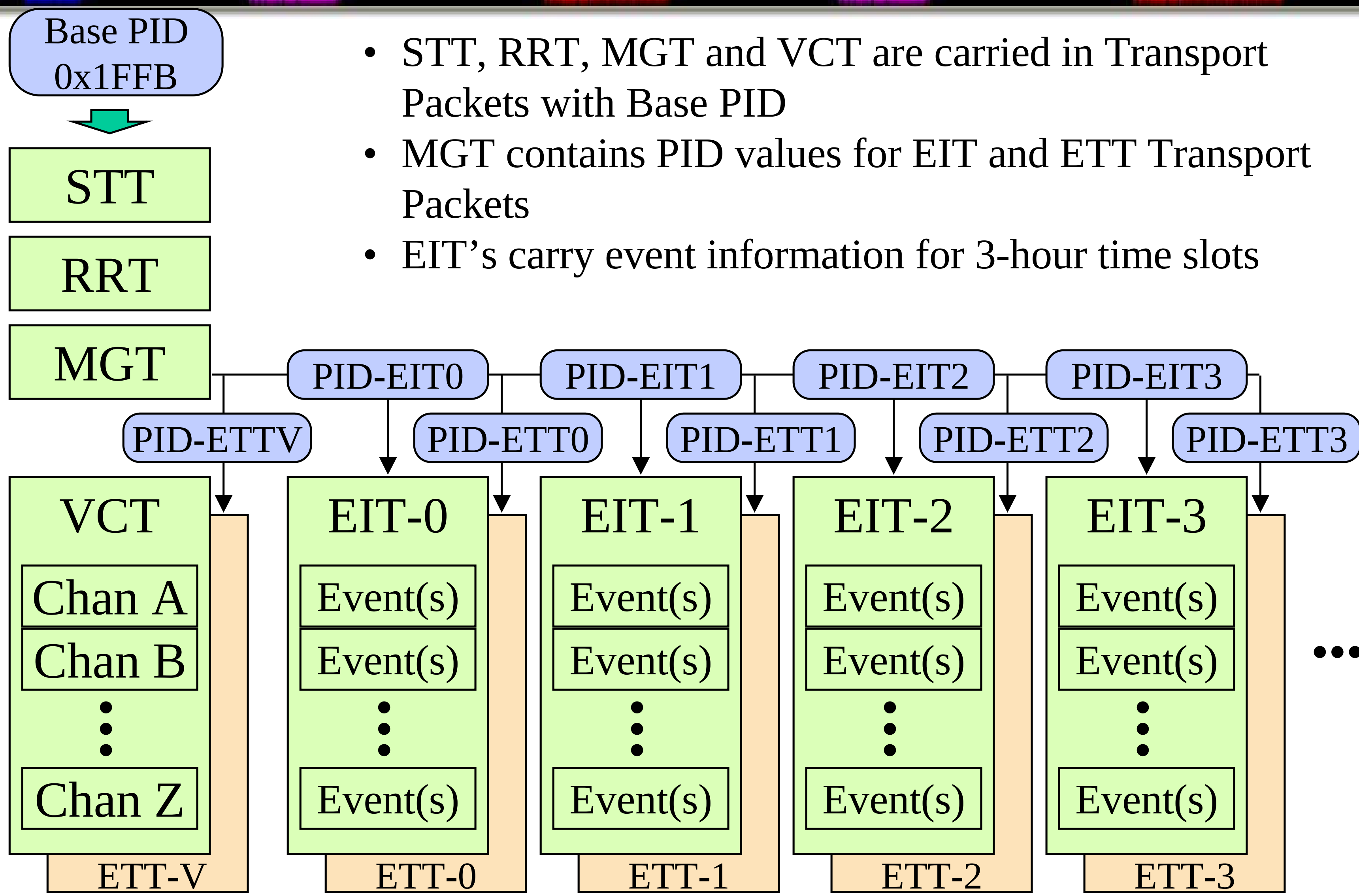
What’s Required for Transmission?

Table	Required for Broadcast?	Required for Cable?
STT	✔	✔
MGT	✔	✔
VCT	✔ (TVCT)	✔ (CVCT)
RRT	✔	✔
EIT	✔ (EIT-0, -1, -2, -3) *(all others optional)*	*optional*
ETT	*optional*	*optional*

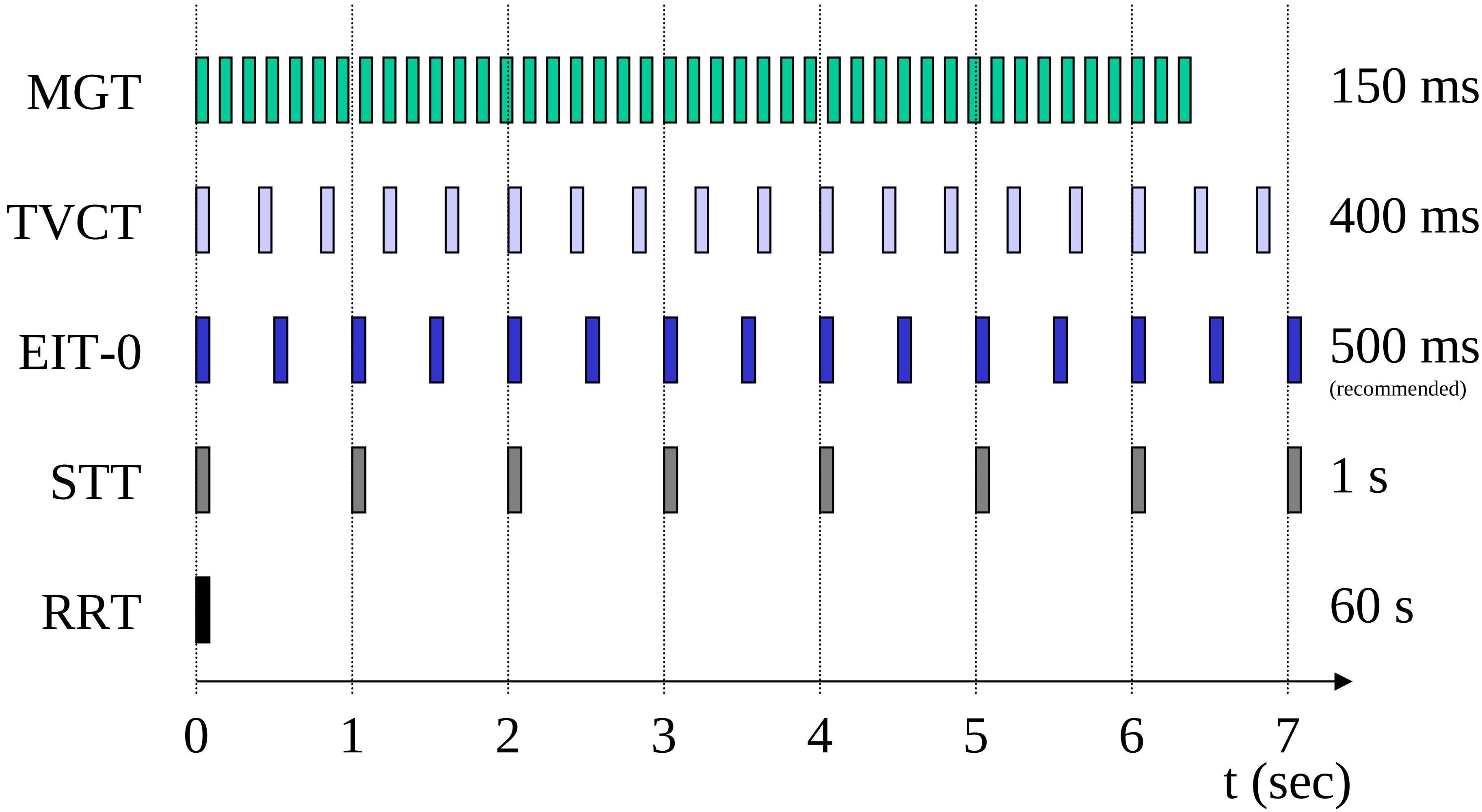
Note: For out-of-band signaling, in cable, refer to SCTE-DVS 234.

Table Hierarchy

- STT, RRT, MGT and VCT are carried in Transport Packets with Base PID
- MGT contains PID values for EIT and ETT Transport Packets
- EIT’s carry event information for 3-hour time slots



Maximum Cycle Times



Master Guide Table (MGT)

- Lists key information about all other PSIP tables (except STT):
  - version numbers
  - table sizes
  - PID's
- Allows simpler decoder designs since any change in PSIP status is flagged in this table.
- Only the base PID (0x1FFB) needs to be monitored to detect change in PSIP status.

MGT Example: Time T_0

MGT
tables_defined = 6; version = 8

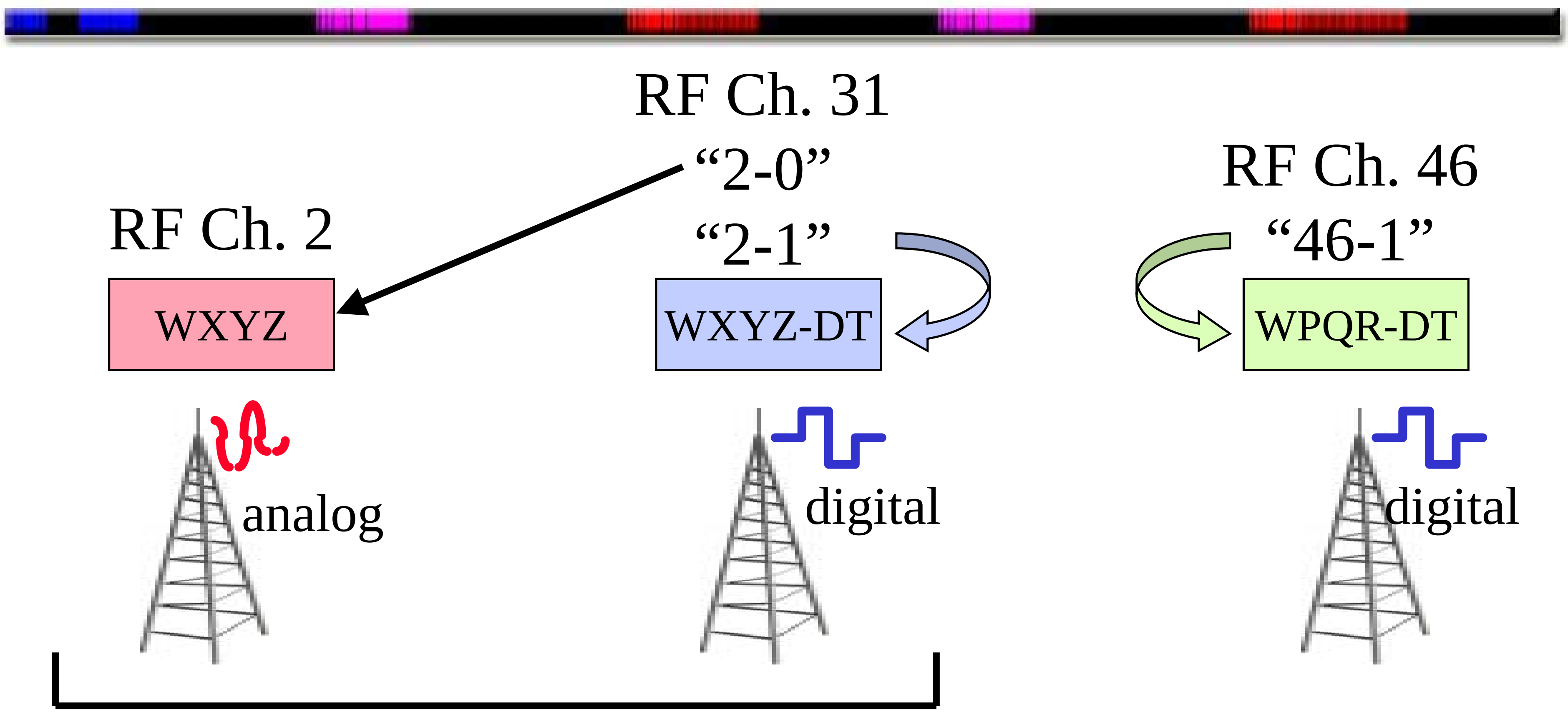
Type	Name	PID	Version	Bytes
0x0000	TVCT (current_next = 1)	0x1FFB	2	450
0x0100	EIT-0	0x1AA0	2	98
0x0101	EIT-1	0x1AA1	2	68
0x0102	EIT-2	0x1AA2	1	77
0x0103	EIT-3	0x1AA3	1	80
0x0301	RRT (rating_region = 1)	0x1FFB	0	990

Note: Underlined values are variable from station to station.

Virtual Channel Table (VCT)

- Contains list of channels in the Transport Stream.
- May also include broadcaster's analog channel and digital channels in other Transport Streams.
- TVCT = Terrestrial VCT; CVCT = Cable VCT
- Key info in VCT:
  - short name
  - major and minor channel numbers
  - Transport Stream ID (TSID) and program number
  - source ID, service type, access controlled and hidden flags
  - Service Location Descriptor: contains list of PID's for elementary streams

Major-Minor Channel Number Example



An existing analog broadcaster with a second digital channel. Branding is preserved. The DTV RF channel number is not needed at all!

A digital-only broadcaster (no analog channel)

PAT and PMT

- The Program Association Table (PAT) associates **MPEG-2 Program Numbers** with Program Map Table (PMT) PID's
- The PMT associates program elements with PID's
- These tables are *required* for MPEG-2 compliance

Program Association Table (PAT)

table_id = 0x00

•••

transport_stream_id = 0x0AA1

•••

Program Number	PMT PID
0x00F1	0x0100
0x00C2	0x0440
0x00B3	0x0301

Example PAT

Program Number

0x00B3

Stream Type	PID
PCR	0x0303
Video	0x0303
Audio	0x0206

Example PMT at PID 0x0301

The Program Number Myth

- MPEG-2 Program Numbers are not related to Major-Minor Channel Numbers!
- MPEG-2 Program Numbers are hidden from the viewer and serve to link MPEG-2 data structures (PAT and PMT).
- Major-Minor channels numbers are what viewers “tune to”!

Terrestrial Virtual Channel Table (TVCT)

Major-Minor Number	Program Number	Channel TSID	Descriptor
12-1	0x00F1	0x0AA1	Service Location
12-2	0x00C2	0x0AA1	Service Location
12-3	0x00B3	0x0AA1	Service Location

What the viewer “tunes to” (Major-Minor Number)

Hidden from the viewer (Program Number, Channel TSID)

Tells the receiver where to find PID’s (Descriptor)

TVCT Example

TVCT
number_channels_in_section = 5; TSID = 0x0AA1

Major Num.	Minor Num.	Short Name	Carrier Freq (MHz)	Channel TSID	Program Number	Service Type	Source ID	Descrip-tors
12	0	NBZ	205.25	0x0AA0	0xFFFF	analog	20	ch name
12	1	NBZ-D	620.31	0x0AA1	0x0F21	digital	21	ch name; serv loc
12	5	NBZ-S	620.31	0x0AA1	0x00B2	digital	38	ch name; serv loc
12	12	NBZ-M	620.31	0x0AA1	0x0CC7	digital	54	ch name; serv loc
12	31	NBZ-H	620.31	0x0AA1	0x0CD0	digital	14	ch name; serv loc

Adapted from A/65

Electronic Program Guides

Chan	Name	6:00 PM	6:30 PM	7:00 PM	7:30 PM	8:00 PM	8:30 PM
6-0	CBZ	City Life		Travel		Movie: *Wild II*	
6-1	CBZ	City Life		Travel		Movie: *Wild II* (HD)	
6-2	CBZ	Movie: *Secret Agent*				Tune 6-1 for Movie: *Wild II* (HD)*	
6-3	LCL	Local News		Airport Info		HD Program on 6-1*	

- Interactive and Useful
  - Event, Channel and Purchase Information
  - Automatic Recording

* With Future Extensions, can enable Thematic Browsing and Sorting- DCC has categories and enables automatic re-direction to retain VC to PID consistency (RFP is out now)

Event Information Tables

- Each EIT spans 3 hours
- Start time for each EIT is constrained to be one of the following UTC times:
  - 0:00 (midnight), 3:00, 6:00, 9:00
  - 12:00 (noon), 15:00, 18:00, 21:00
- EIT-0 represents the ‘current’ 3 hours of programming
- For terrestrial PSIP, first 4 EIT’s (EIT-0, -1, -2, -3), representing 9 to 12 hours, are <u>required</u>
- Maximum number of EIT’s = 128 (16 days)

EIT Example

EIT-0
source_id = 22
num_events_in _section = 3

Event ID	Local Start Time	Length (seconds)	ETM Location	Title	Descrip-tors
51	12:30	7200	01 (this PTC)	Soccer Live	content_ advisory
52	14:30	3600	00 (no ETM)	Golf Report	closed_ caption
53	15:30	9000	01 (this PTC)	Car Racing	content_ advisory

Adapted from A/65

PSIP and Data Services (A/90)

- Announcement via Extension to PSIP
  - Data Event Table (DET) - based on EIT
    - For Separate Data Services
    - Points to the VCT
    - VCT Points to new Structure
- Announcement via EIT
  - Data Information may be in EITs
    - For Related Data Services
- Binding via new Table Structures
  - Service Description Framework

Relevant PSIP Documents

- PSIP Standard (A/65A)
- PSIP Amendment 1 to A/65A (Directed Channel Change)
  - *ATSC T3 re-ballot just completed*
- Conditional Access System for Terrestrial Broadcast (A/70)
  - Defines ATSC_CA_descriptor for VCT and EIT
- "U.S. Region Rating Table (RRT) and Content Advisory Descriptor for Transport of Content Advisory Information Using ATSC A/65 Program and System Information Protocol (PSIP)", September 1998 (EIA-766)
  - Used for rating and content advisory in the U.S.

THE ATSC DATA BROADCAST SPECIFICATION: PRACTICAL IMPLEMENTATION CONSIDERATIONS

Regis J. Crinon, Ph.D.
regis.j.crinon@intel.com

Intel Corporation
JF3-206
2111 N.E. 25th Avenue
Hillsboro OR 97124

OUTLINE

1. DTV Data Services: Generalities
2. The A/90 specification
   - composition
   - scope
3. Data Service schedules announcement
   - types of services
   - extensions to PSIP
4. Protocols
5. Service Description Framework
6. Data Service profiles and levels
7. Summary and conclusion

DTV = Audio + Video + Data

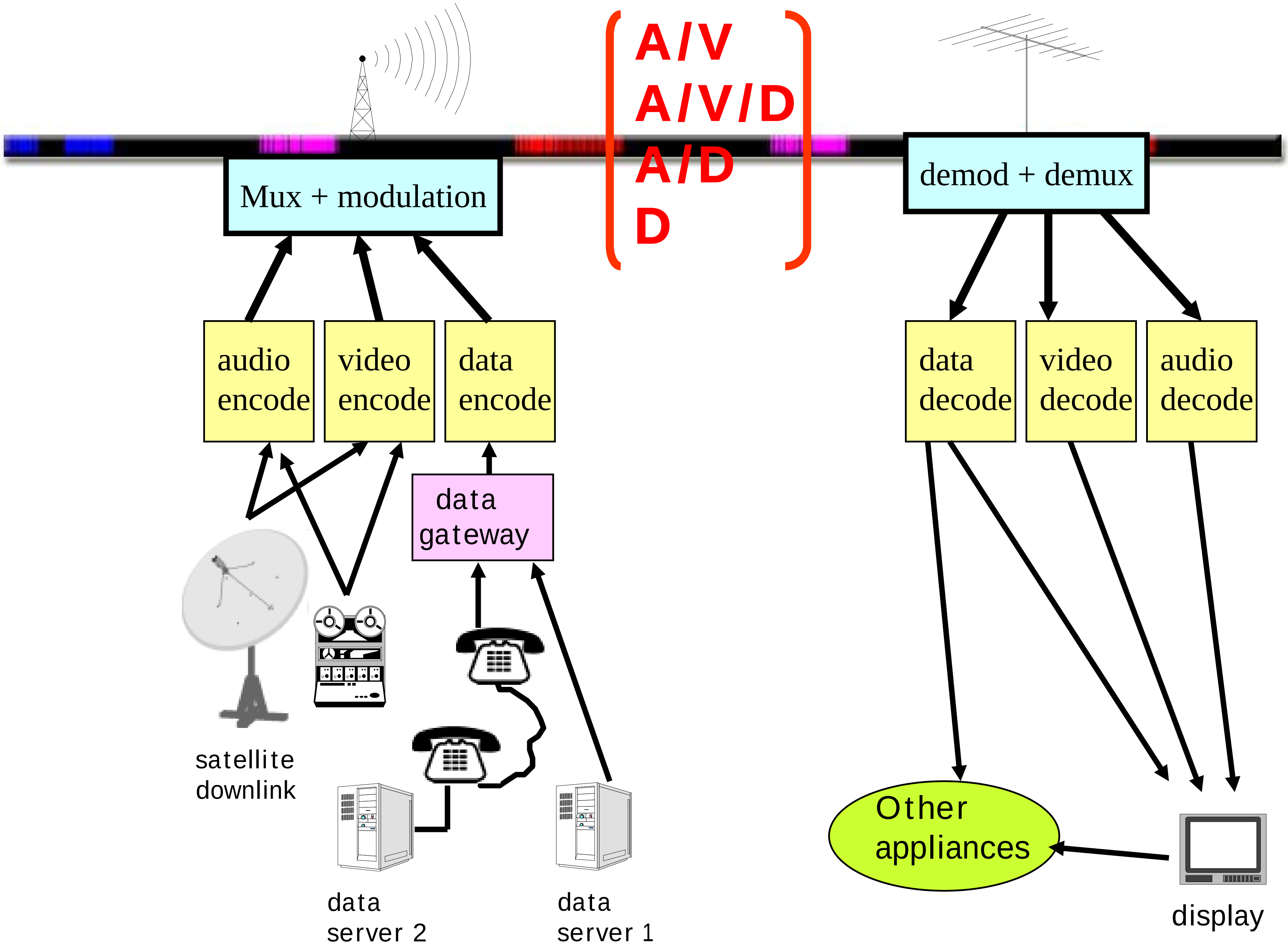
ATSC Transport Stream carrying multiplexed:

- AC-3 audio streams (A/53)
- MPEG-2 video streams (A/52)
- Service Information (A/65 + MPEG-2 SI)
- *data elementary streams (A/90)*

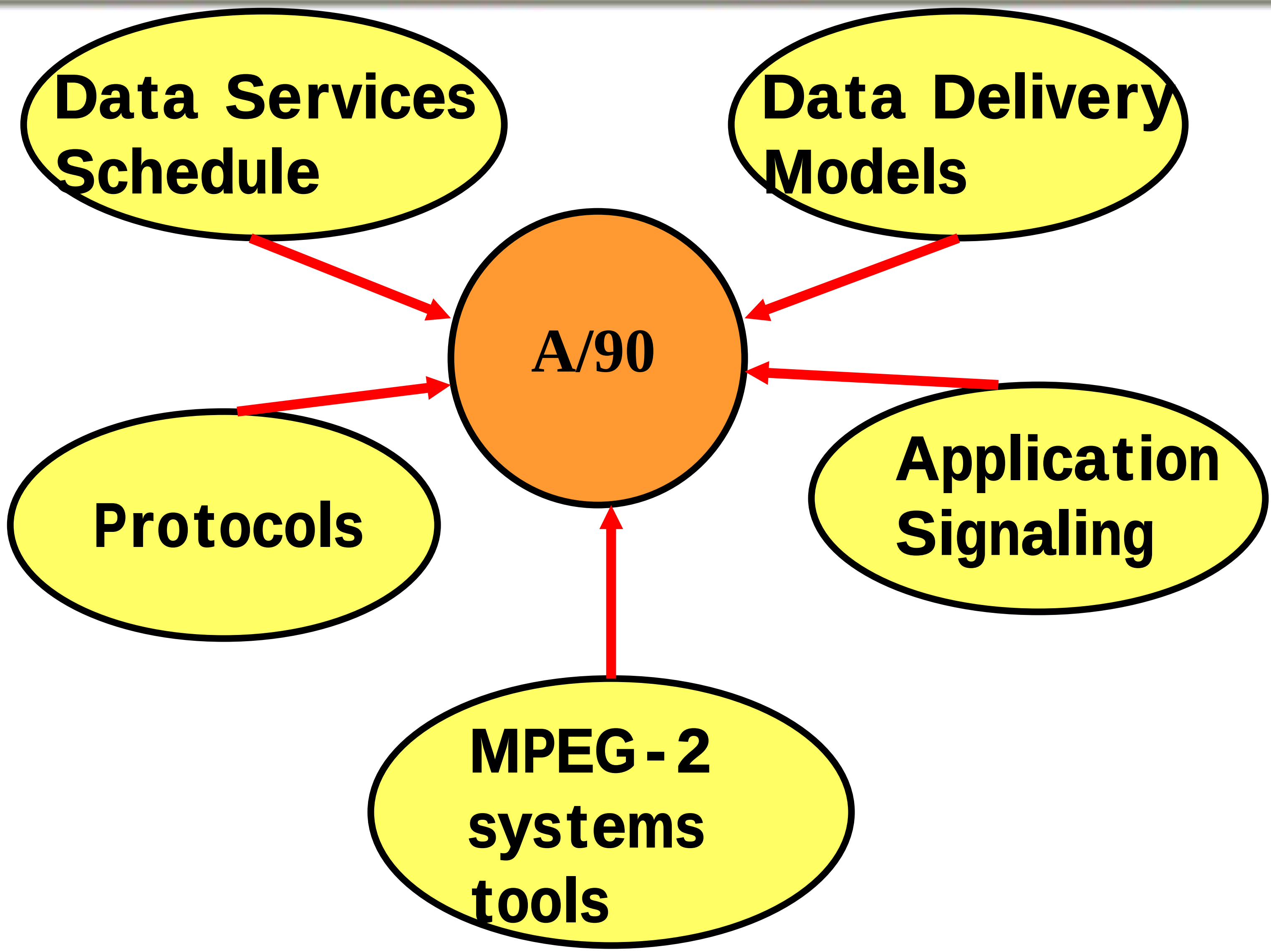
IMPACT ON DTV MARKETABILITY

As opposed to NTSC VBI-based Data Services, DTV Data Services are an integral part of the Broadcast signal:

- Data share the same multiplex with video and audio
- Same fundamental MPEG-2 acquisition mechanisms are used to acquire data, video and audio.
- Data Services may be announced in a Program Guide like Video/Audio programming

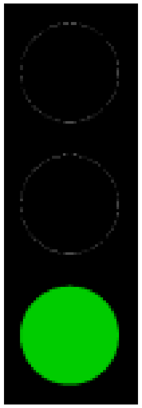

A/V
A/V/D
A/D
D
Mux + modulation
demod + demux
audio encode
video encode
data encode
data gateway
data decode
video decode
audio decode
satellite downlink
data server 2
data server 1
Other appliances
display

WHAT DOES ATSC A/90 SPECIFY ?



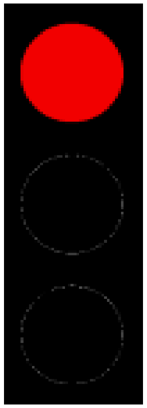
SCOPE OF ATSC A/90

Examples of what it can be used for :



- Delivery of declarative data, Java code
- Delivery of software, images, graphics
- MPEG-4 or H.263 video streams (data piping)
- MPEG-4 audio streams (data piping)
- Carousel of MPEG-2 video files (.mpg files)
- Carousel of MP3 audio files

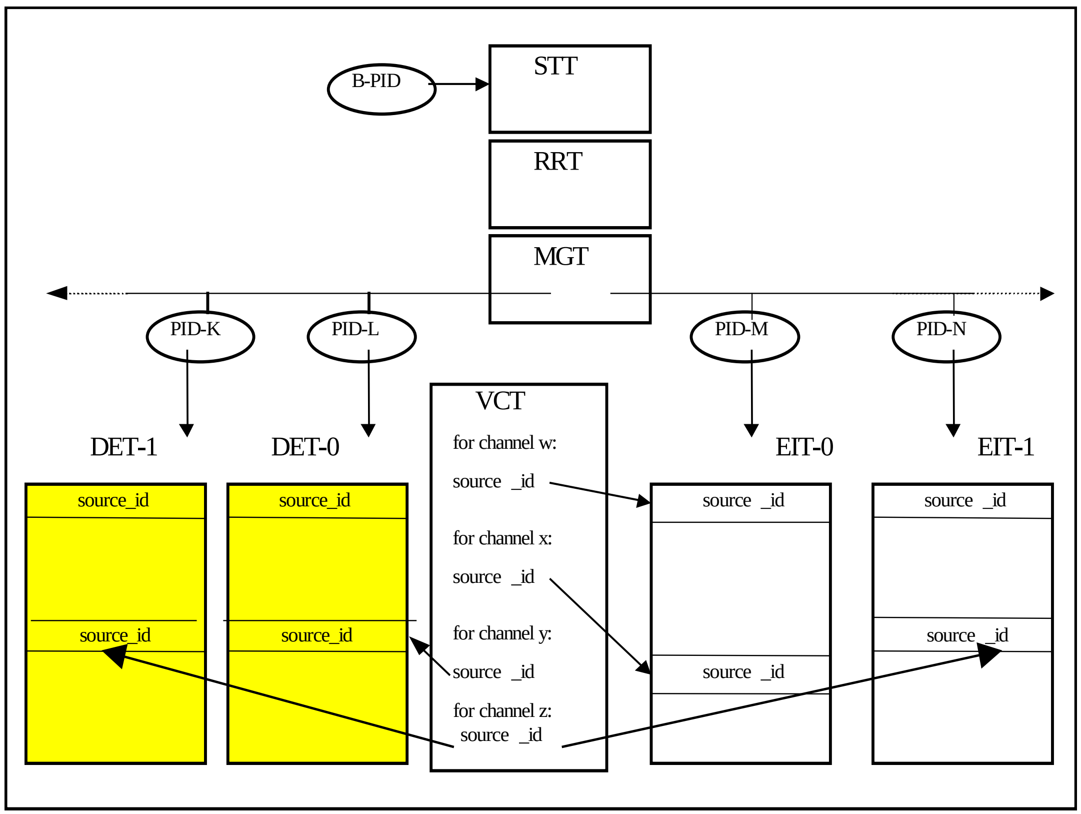
What it cannot be used for :



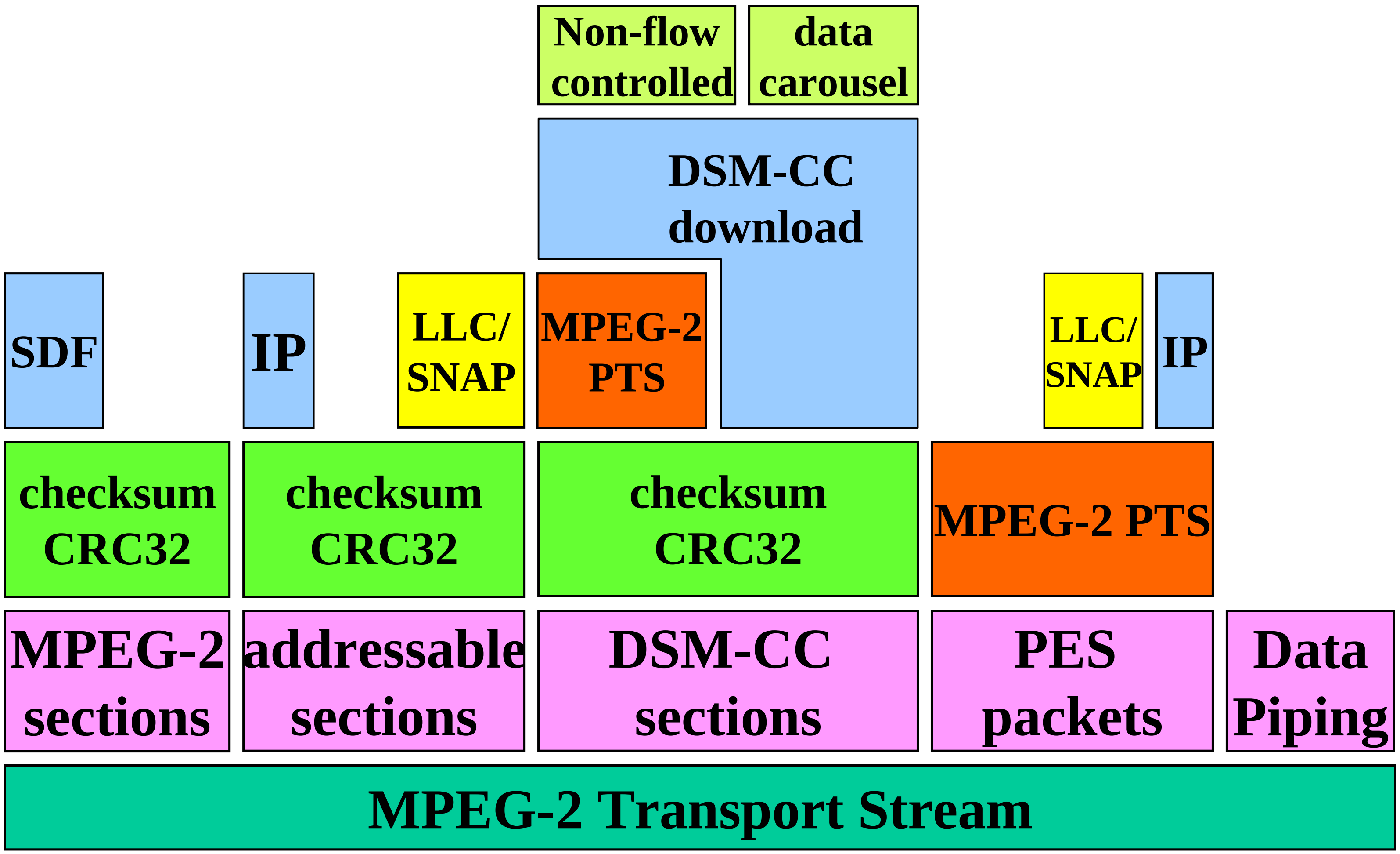
- Audio elementary streams of type 0x81
- Video elementary streams of type 0x02

ANNOUNCEMENTS OF DATA SERVICE SCHEDULES

- One data service per virtual channel !
- A/90 has invented DET-k's

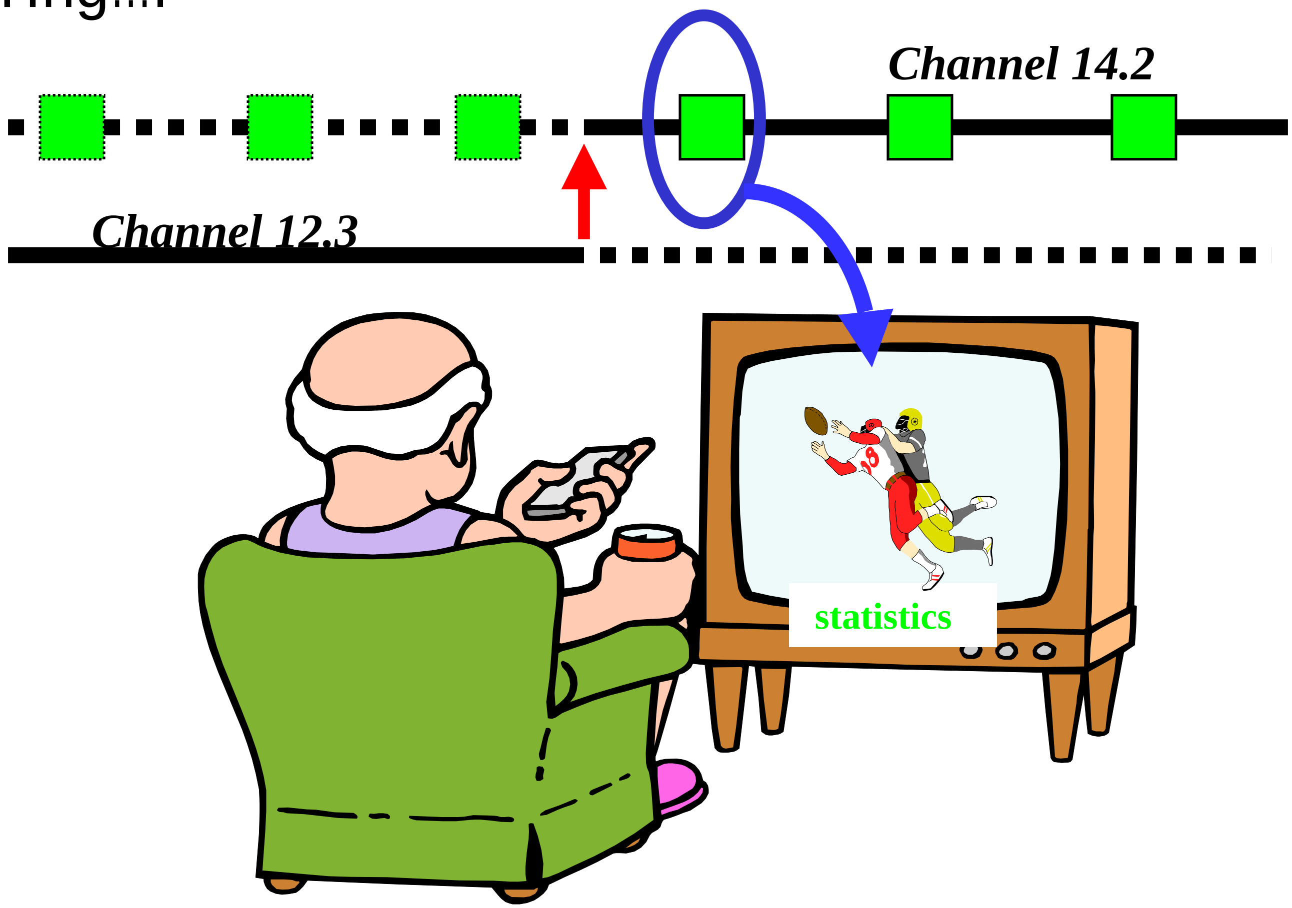


PACKETIZATION, ERROR PROTECTION AND PROTOCOLS



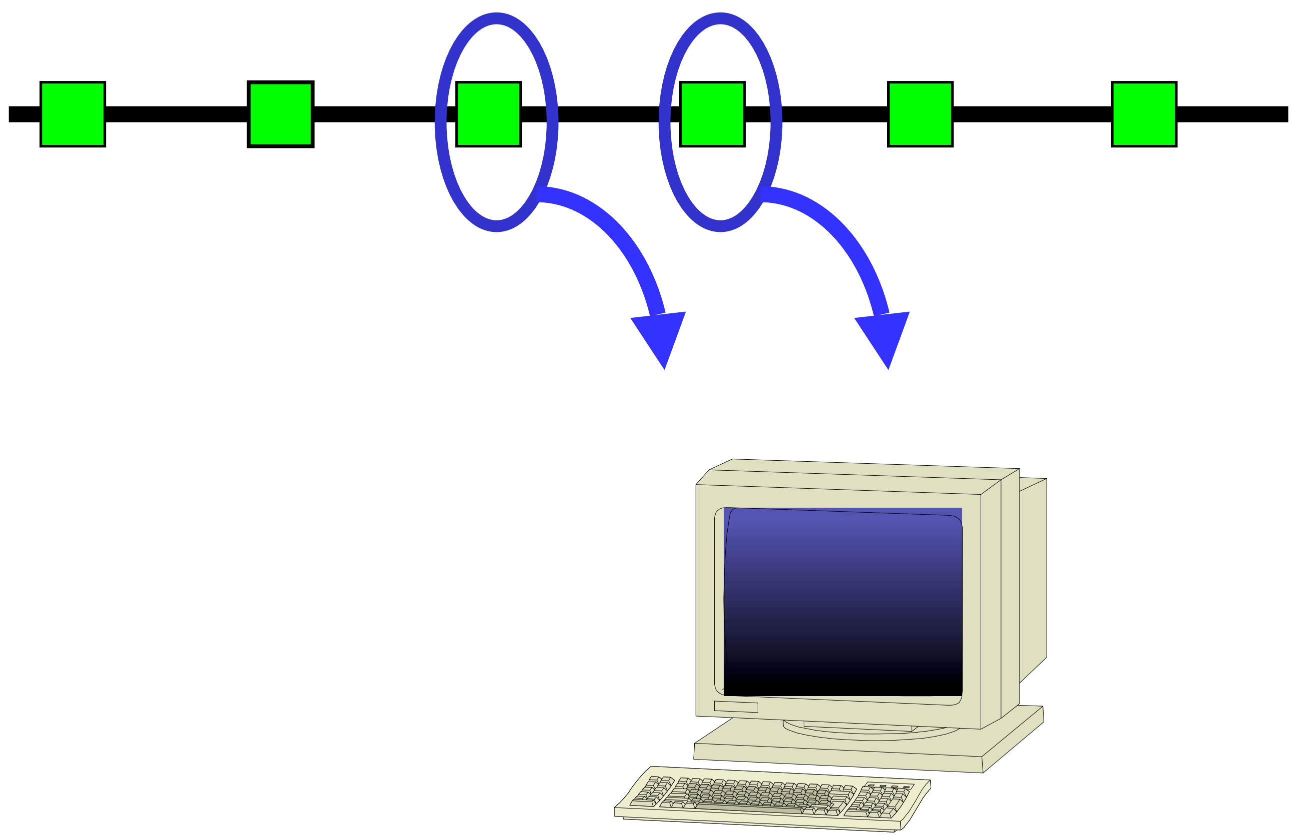
DSM-CC DATA CAROUSEL

Periodic re-transmission of the same data to allow content providers to cope with viewers channel surfing....

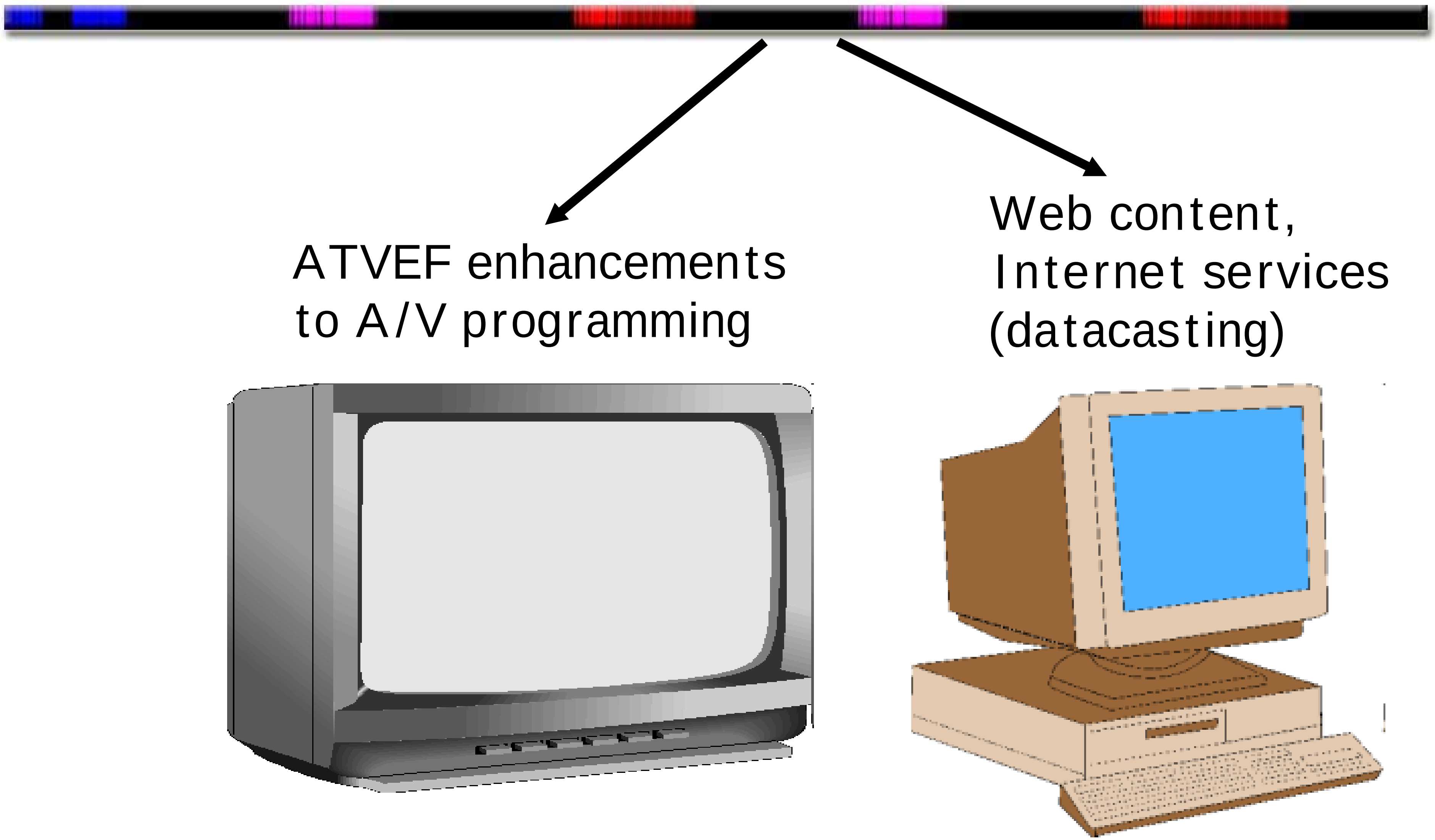


DSM-CC DATA CAROUSEL

Periodic re-transmission of the same data to allow content providers to cope with loss of data...
Example: Software download.

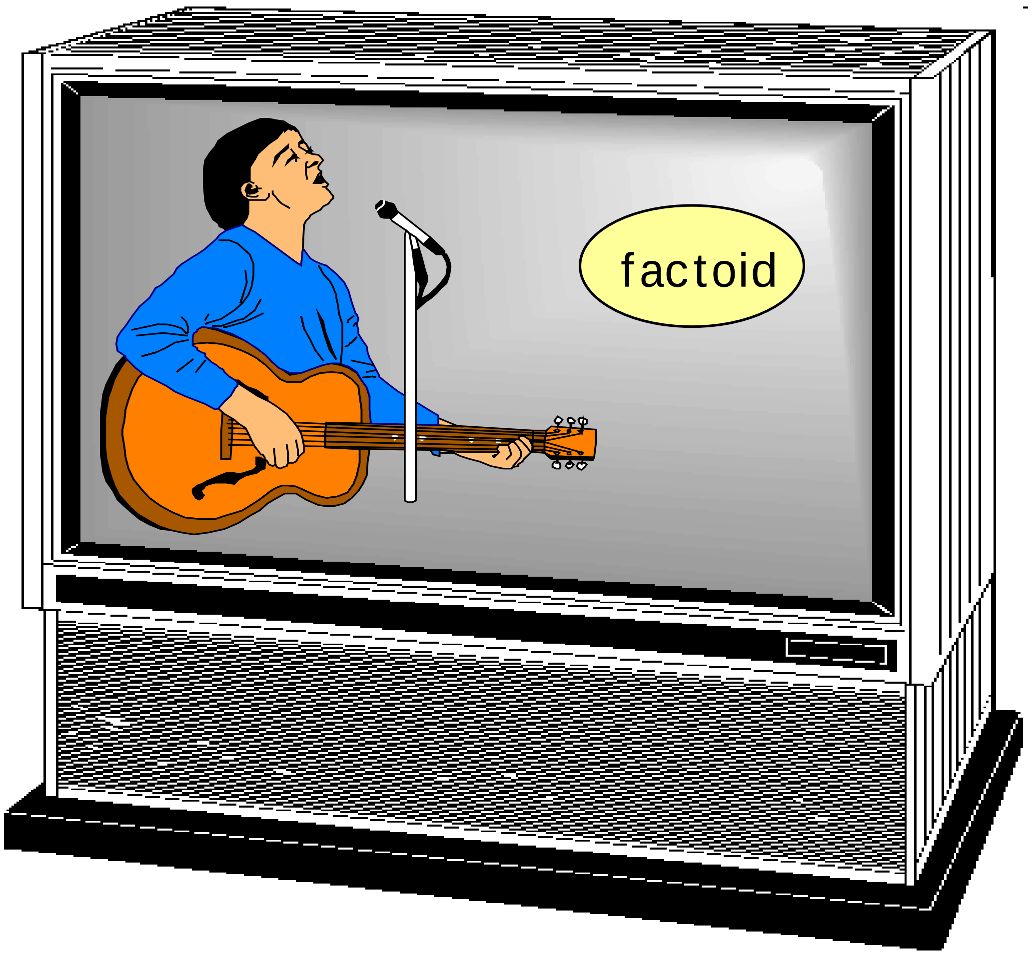


ADDRESSABLE SECTIONS CARRYING IP DATAGRAMS



SYNCHRONIZED DATA

Data is synchronized with audio/video content



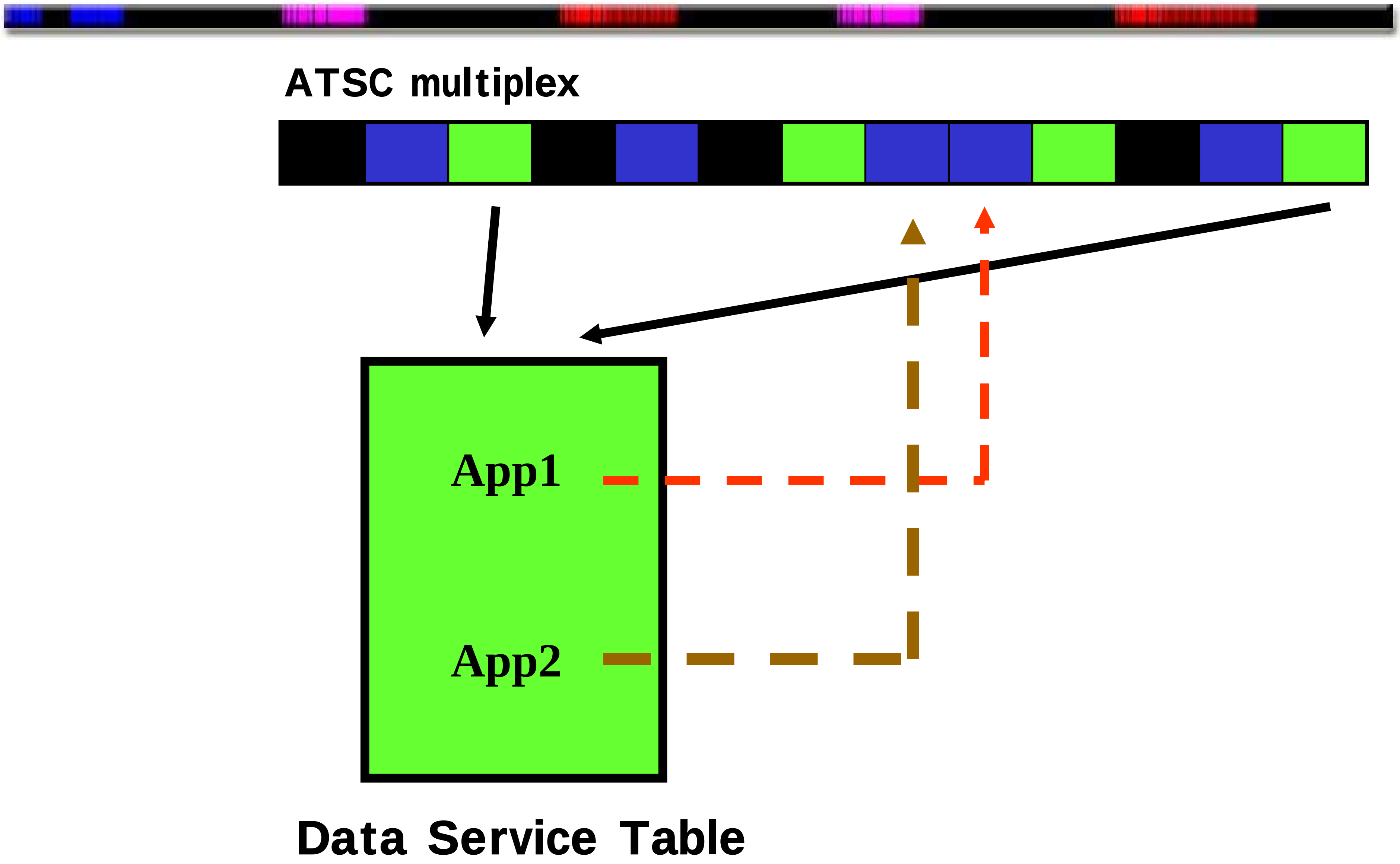
APPLICATION SIGNALING

ATSC Virtual Channel

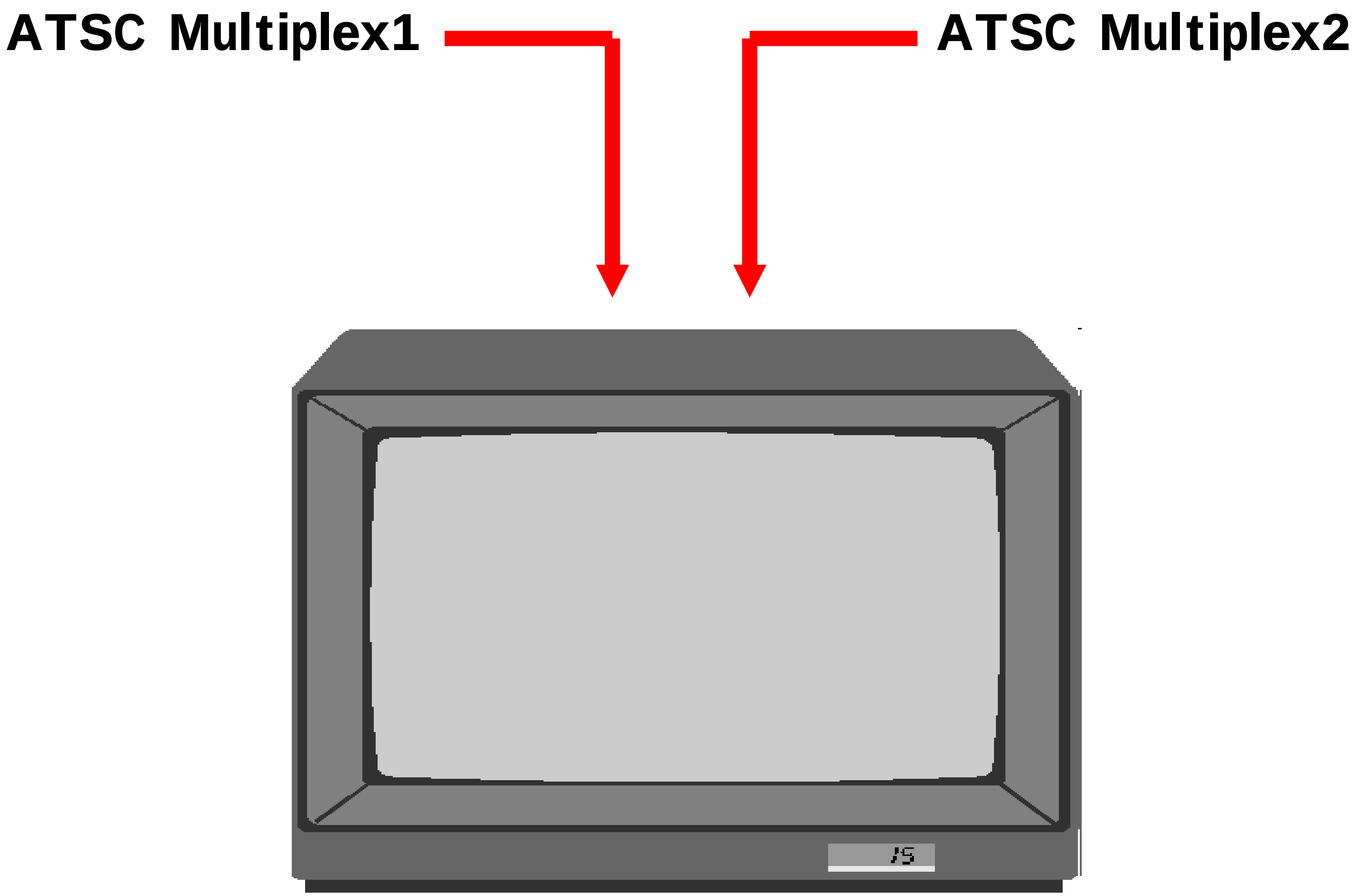
- Video elementary stream
- Audio elementary stream
- Data elementary stream 1
- Data elementary stream 2
- Data elementary stream N
- SDF data
  - Data Service Table
  - Network Service Table

SDF data for discovery and binding of the data components used by a receiver application.

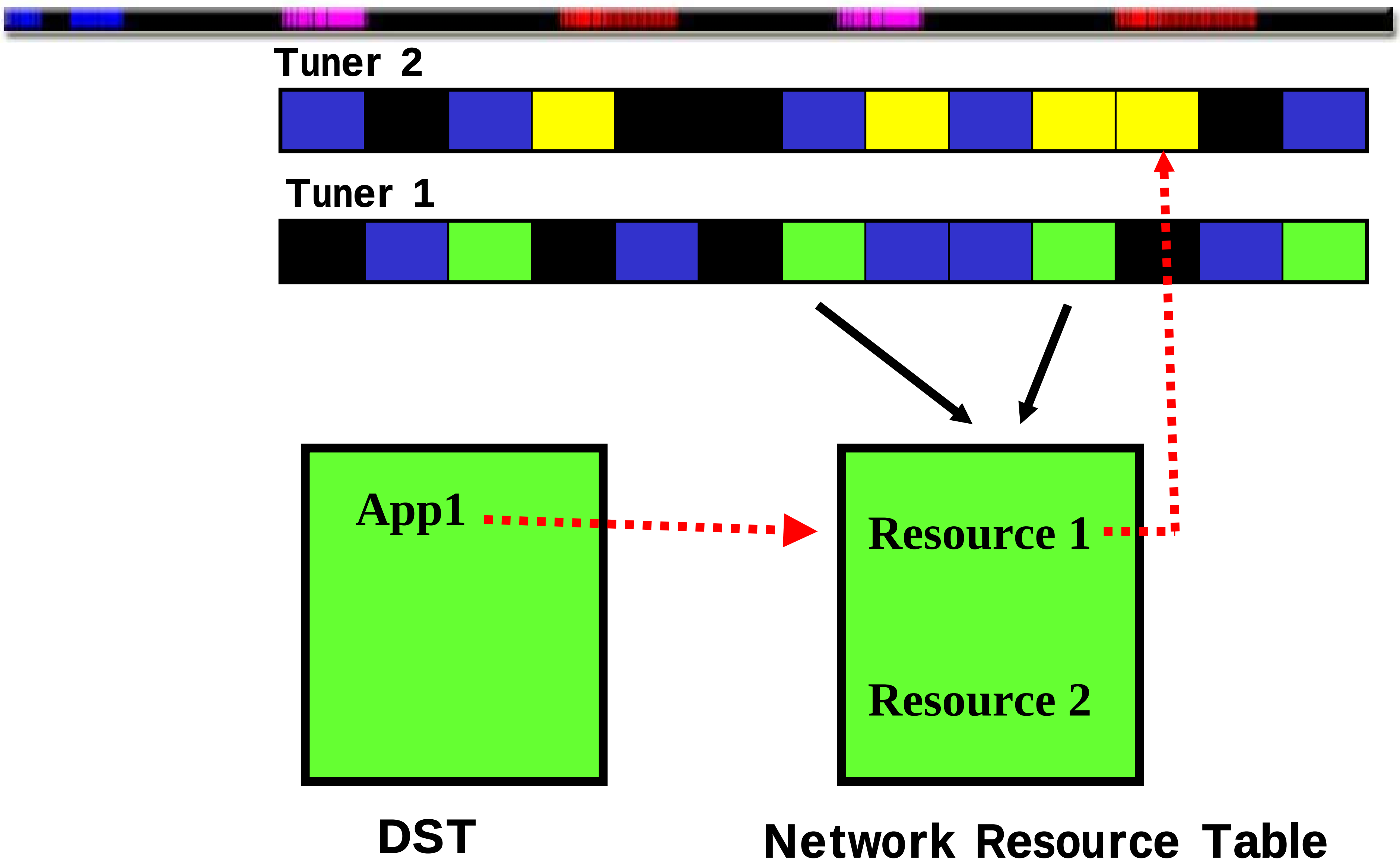
REFERENCE TO DATA WITHIN THE SAME VIRTUAL CHANNEL



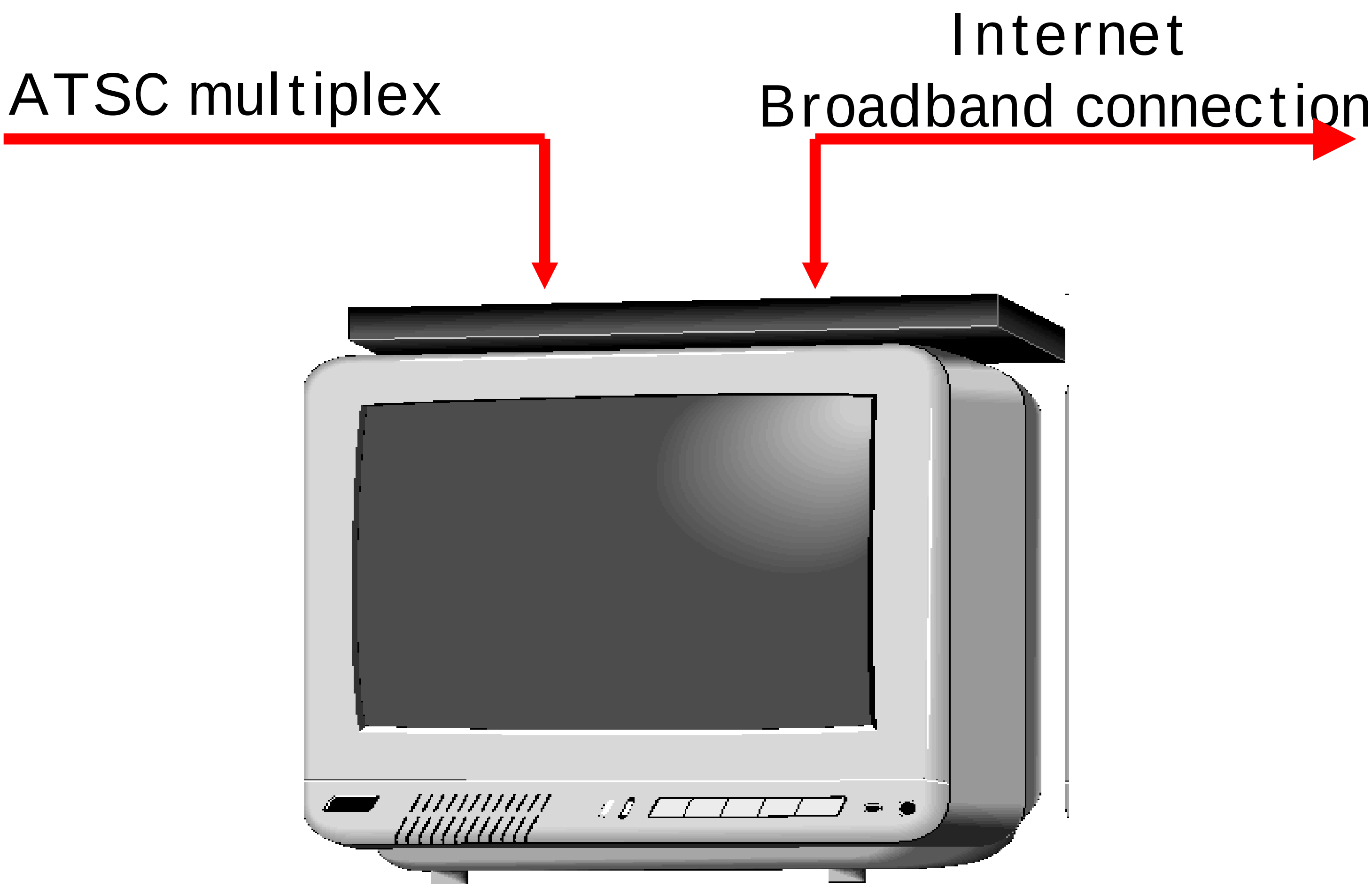
DATA SERVICES FOR DUAL-TUNER RECEIVERS



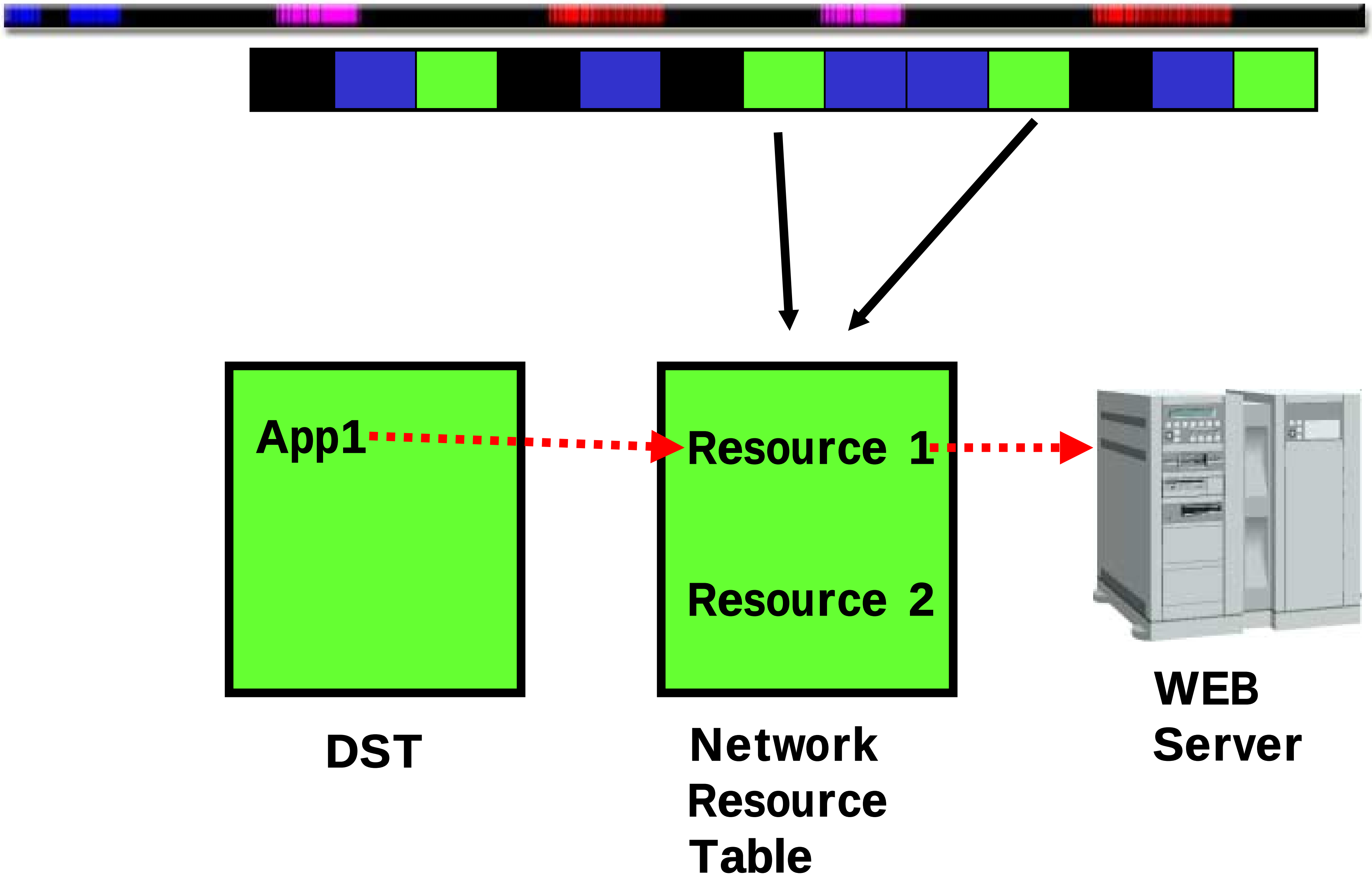
REFERENCE TO DATA IN ANOTHER VIRTUAL CHANNEL



DATA SERVICES FOR RECEIVERS FEATURING A BROADBAND CONNECTION



REFERENCE TO DATA AT A REMOTE WEB SITE



APPLICATION SIGNALING

SDF data is part of the data service so content providers must provision (and pay) for its transmission

Content provider may select how often and how big SDF information can be.

Trade-off between:

- tune-ability to data services
- latency to get access to data

DATA SERVICE PROFILES AND LEVELS

- Signaled in a descriptor carried in the EIT-k's or DET-k's
- Profiles determine the maximum bitrate that a data service consumes
- Levels are linked to receiver memory and throughput requirements for synchronized services

FOUR DATA SERVICE PROFILES

G1	Guaranteed bandwidth up to 384 kpbs
G2	Guaranteed bandwidth up to 3.84 Mbps
G3	Guaranteed bandwidth up to 19.2 Mbps
A1	Opportunistic up to 19.2 Mbps

(NTSC VBI-based Data Services: **180 kbits/sec** max)

DATA SERVICE PROFILES

Two classes of profiles:

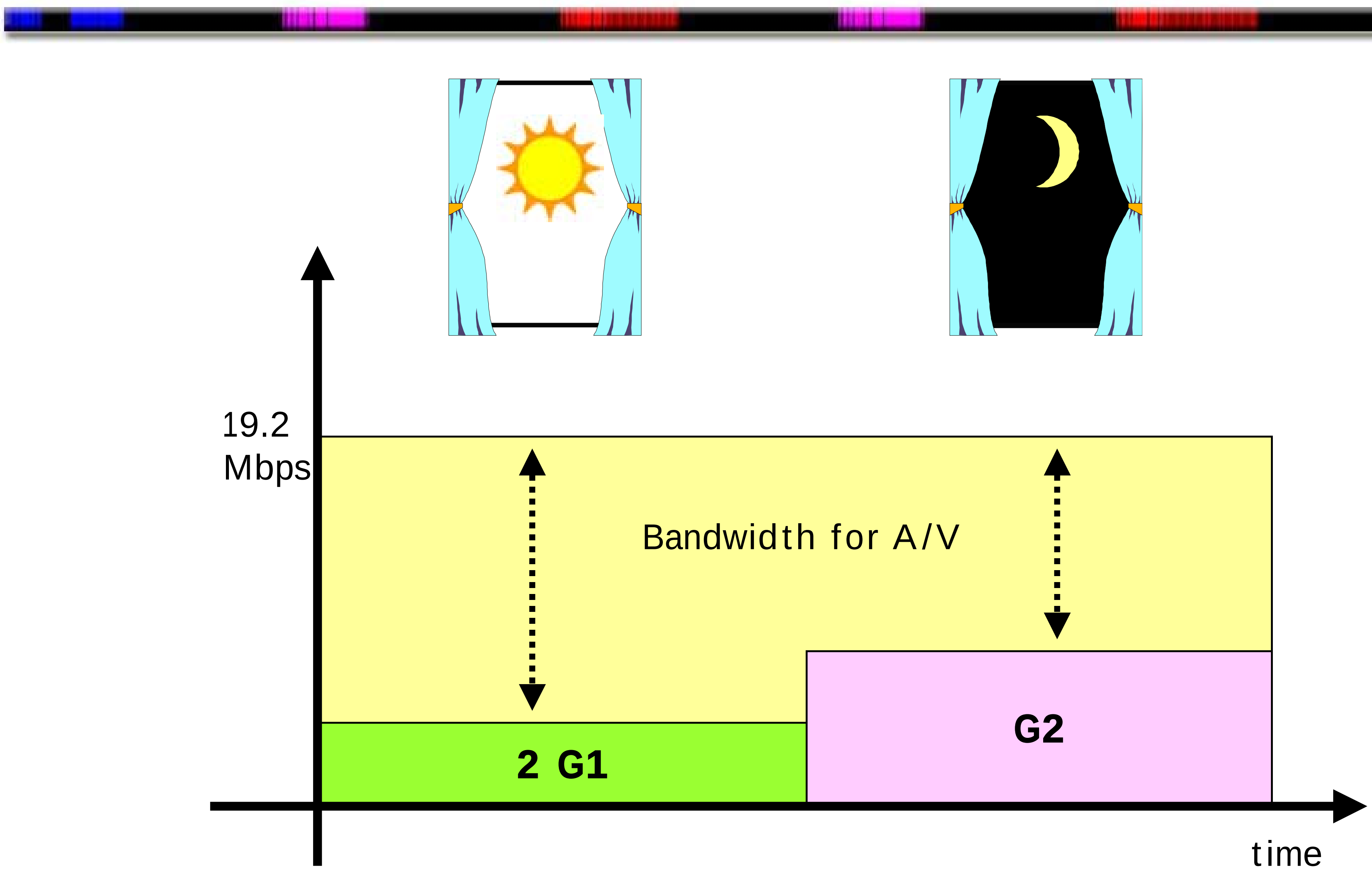
- Guaranteed bandwidth

  Specifies maximum bandwidth that has been provisioned for transmission of data service.

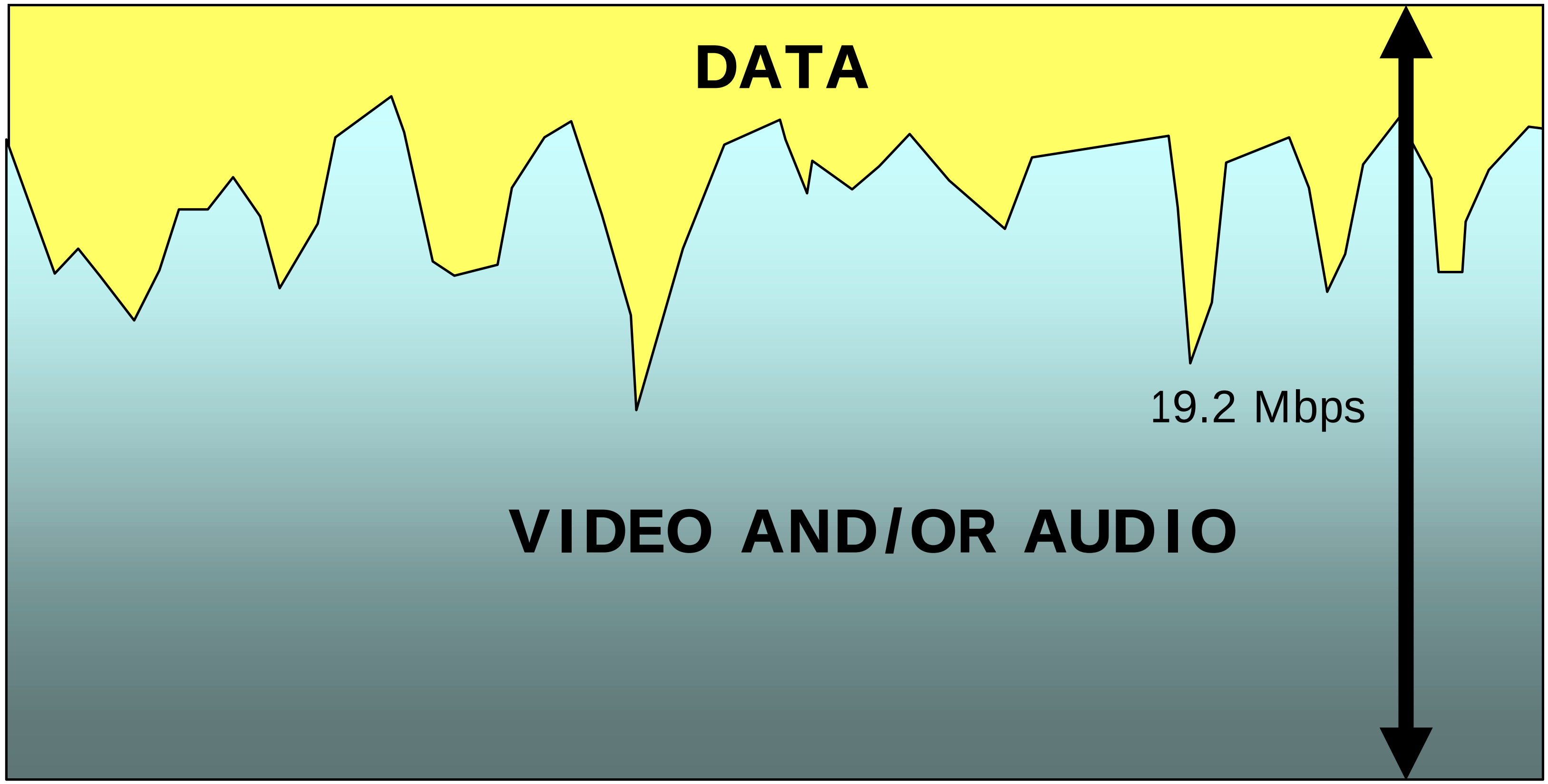
- Opportunistic bandwidth

  Bandwidth assigned to transmission of data service is variable in time, depending on *instantaneous* availability (bandwidth not used by audio and video)

GUARANTEED BANDWIDTH



OPPORTUNISTIC BANDWIDTH



USE OF DATA SERVICE PROFILES

At the head-end

Data Service profiles enable brokerage of total bandwidth reserved for data services in an ATSC multiplex:

G3 = 5 G2 = 50 G1 = 4 G2 + 10 G1 = ...
5.76 Mbps = G2 + 5 G1

At the receiver

Data Service profiles specify targeted receiver capability

FOUR DATA SERVICE LEVELS

Level1	DEBSn = 120120 bytes
Level4	DEBSn = 480480 bytes
level16	DEBSn = 1921920 bytes
Level64	DEBSn = 7687680 bytes

Max throughput at level 1 = 172.8 Mbits/sec

SUMMARY

- Rich set of protocols and functionalities will allow progressive deployment of increasingly more sophisticated services

- A/90 specifies delivery data for broadcast and pseudo-interactive services but at the same time lays the ground for interactive services.

- A/90 was input to SCTE for review (DVS161)
- A/90 has a high level of compatibility with DVB:
  1) Data Carousel
  2) Synchronized protocol
  3) Carriage of IP datagrams
- Led to 3 MPEG Systems amendments

THE ATSC T3/S13 DATA BROADCAST SPECIALIST GROUP

The Data Broadcast Standard is available on the ATSC Web Site:
http://www.atsc.org/Standards/A90/A90.pdf

A companion implementation guide is available in the form of a recommended practice

Working on a IP Multicast specification

THE ATSC IS / DIWG
Data Implementation Working Group

DIWG report is available on the ATSC Web Site:

http://www.atsc.org/Standards/IS_151.pdf

Everything you ever wanted to know about Data Broadcast

- Data Broadcasting: Understanding the ATSC Data Broadcast Standard
  - R. Chernock, R. Crinon, M. Dolan, J. Mick
  - McGraw Hill, 2001

Application Reference Model

Michael A. Dolan

Industry Consultant,
miked@tbt.com

ATSC is working on data broadcasting transport issues (A/90) as well as data-specific application environments (DASE). However, when it comes time to implement a receiver, there is a gap between them where normative bindings and behavior needs definition. This is the Application Reference Model (ARM). It covers a uniform naming system, data model characterization, and an application state model. This coverage addresses such things as defining MPEG descriptors to provide the proper name bindings, and provide guidance on state transitions based on events in the transport signaling. Also included are data models such as files, streams, and IP packets. In summary, this links together all the basic constructs in the ATSC data transport to ATSC-DASE, and is general enough to be used by other application environments if needed.

NIST/ATSC Symposium: End-to-End Data Services, Interoperability and Applications

Data Application Reference Model

Michael A. Dolan

19-June-2001

Overview

- Provide the glue between A/90 transport and DASE application environment
- Application model builds on several ATSC standards
- Main areas of focus are:
  - Naming system
  - Data Model Characterization
- Application State Model

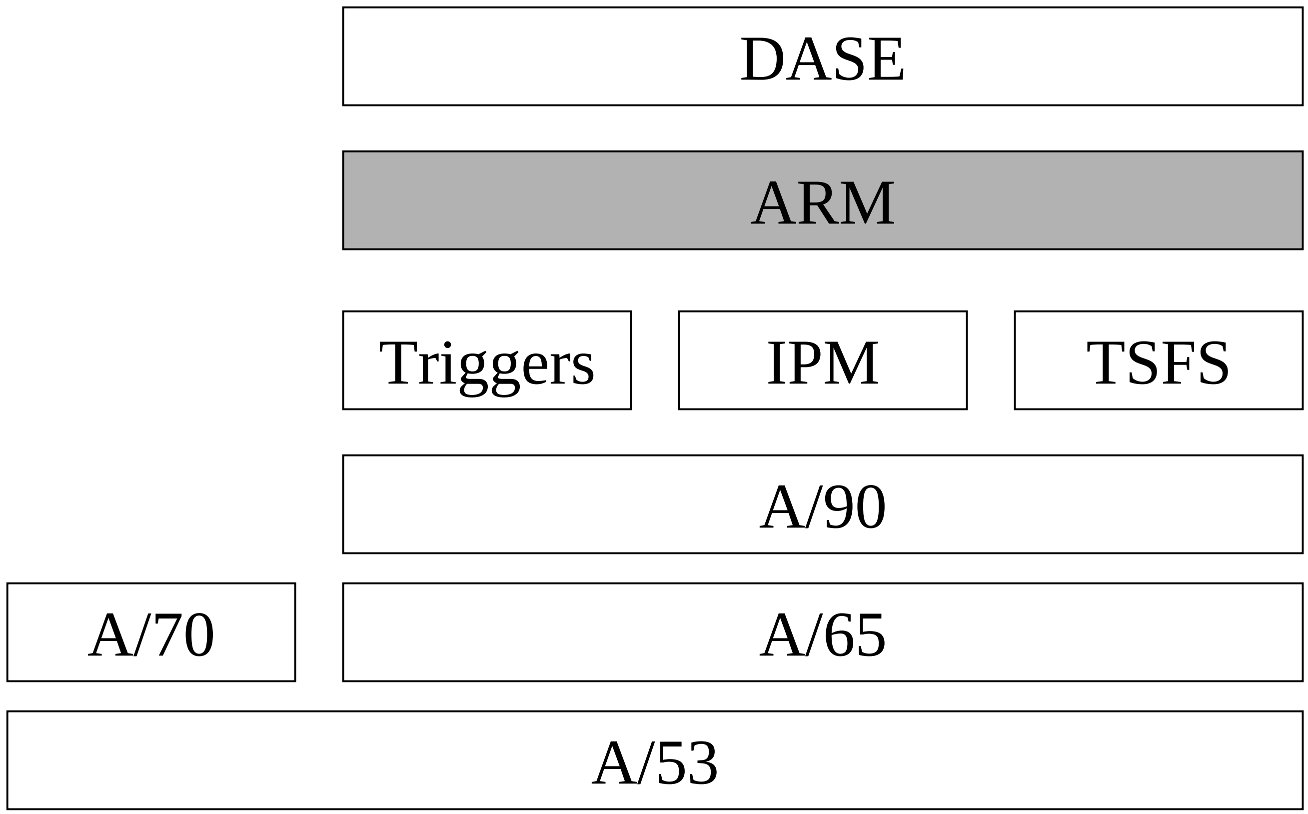
Warning & Disclaimer

- This presentation discusses ATSC work in process, and therefore cannot be relied on for product development or even excepted final endorsement by the ATSC. This is an informative presentation about current thinking of the technical experts on a topic relevant to this audience.

ATSC Standards Relationship

– A/53 (Core ATSC video & audio)
– A/70 (CA)
– A/65 (PSIP)
– A/90 (Data Broadcast Framework)
– S13 Work in Process
  • IP Multicast (IPM)
  • Triggers
  • Transport Stream File System (TSFS)

ATSC Standards Relationship



Data Models

- Modules
- Files
- Streams
- IP Packets
- Triggers

Module Data Model

- Similar to files, but generally only used for:
  - Receiver firmware upgrades
  - Synchronized downloads
  - Limited naming scenarios
  - Other “simple” scenarios
- (no mapping to DASE data model)

File Data Model

- Bounded sequence of bytes
- Just like a computer file system
- Hierarchical namespace with directories
- Carried in DSMCC modules
- Defined by T3/S13 in TSFS Standard
  - (Will likely be Object Carousel)
- Maps to DASE resource

Stream Data Model

- Unbounded sequence of bytes
- Like a UNIX pipe or IP/TCP connection
- Carried in either:
  – DSMCC Asynchronous Download
  – Data Piping
- Defined by application
- Maps to DASE JMF built-in data sources

IP Packet Data Model

- Internet Protocol Packets
- Primarily Multicast only
- Carried in DSMCC Addressable Sections
- Defined in T3/S13 IP Multicast Standard
- Maps to DASE datagram socket

Trigger Data Model

- Event delivery to receiver
- Supports both targets:
  - Synchronized Module
  - Application Event
- Carried in DSMCC Download
- Defined by T3/S13 Trigger Standard
- Application Event maps to DASE DOM Events

Naming System

- Need to provide names for the transport resources
- Each data model is supported
- tv: URI scheme used for current video/audio
  - RFC 2838
- lid: URI scheme used for all other resources
  - SMPTE work in process
- Signaling is via descriptors in the DST, as well as DII (for modules)

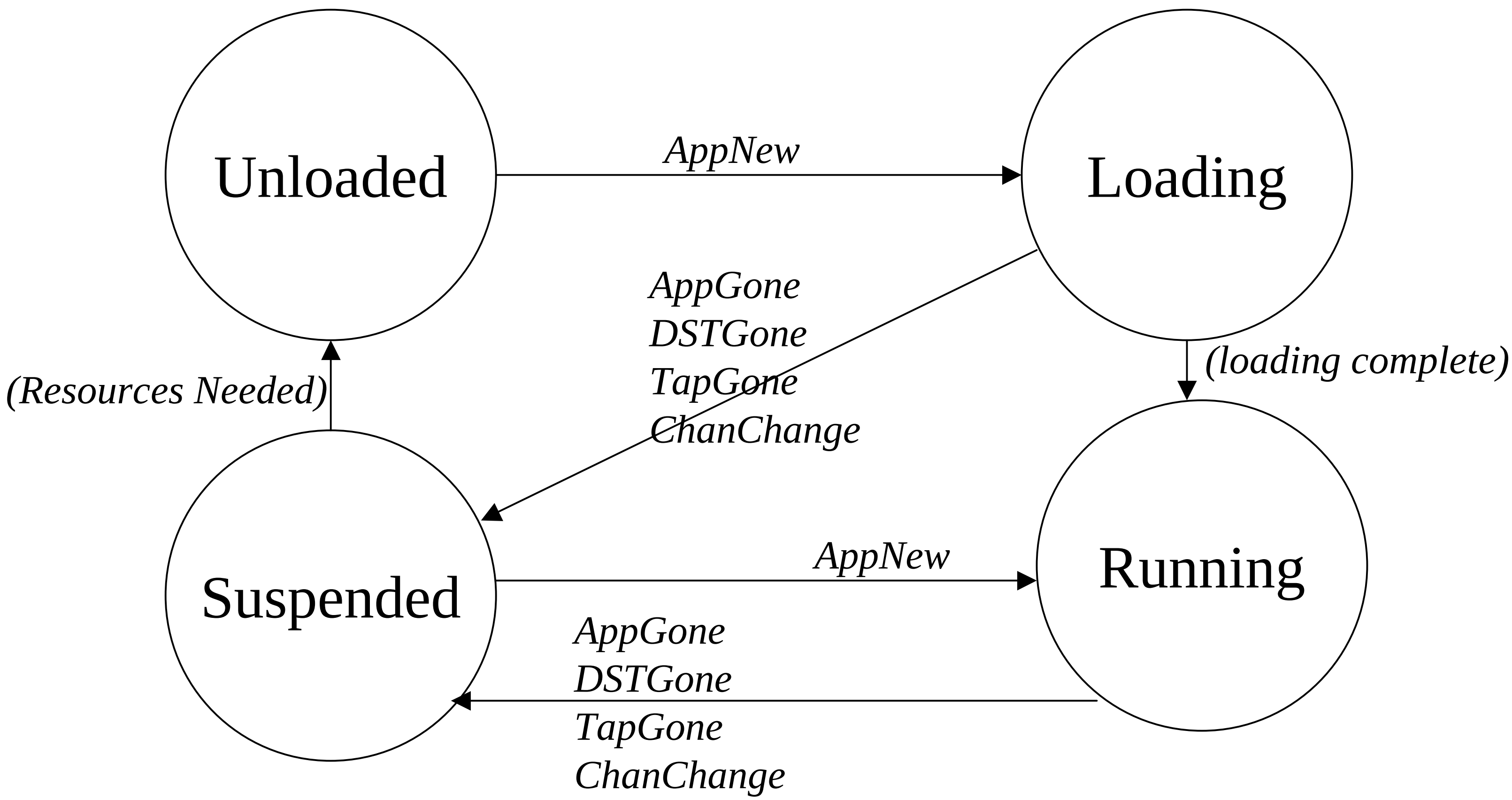
State Model

- Needed to provide basic transport layer environment management
- Based on A/53, A/65 and A/90 signaling
- Input events are existing transport signals
- States are abstract

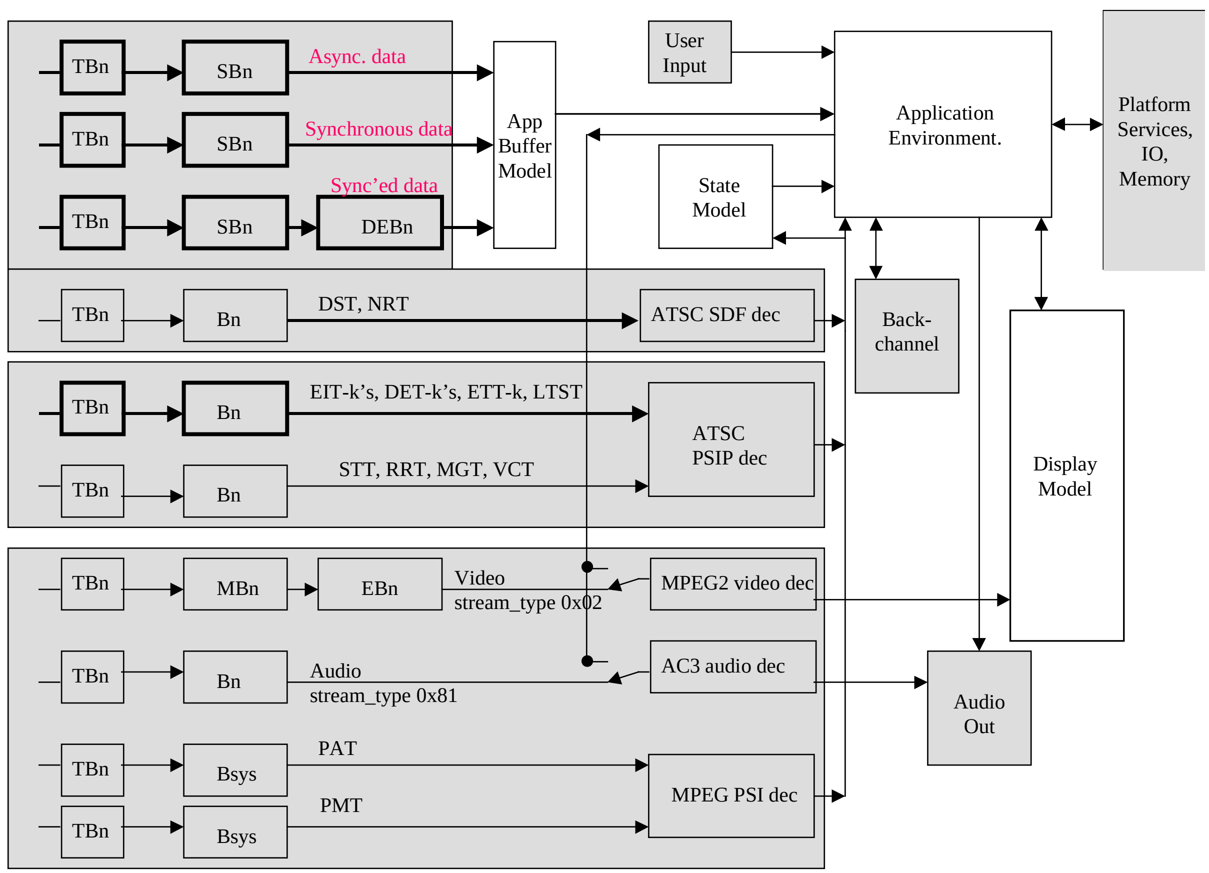
State Model *Events*

- DST contains a new application
- DST omits a previous application
- PMT omits the DST
- PMT omits the Program Element that contains the “boot” resource
- Channel Change

State Transition Diagram



Receiver Block Diagram



A/90 Extensions & Constraints

- New Information
- Announcement
- Signaling
- Encapsulations

New Information

- appID = UUID
- Compatibility Descriptor
- Identifiers (lid:)
- Content Type
- Broadcaster Permissions

Compatibility Descriptor

- Organization (OUI)
- Capability
- Profile
- Level

Content Type

- “MIME” Type
- More clearly defines content of:
  - Modules
  - Files
  - Streams
  - Triggers

Broadcaster Permissions

- High level broadcaster control
- Permits denial of application functionality
- Usable by data service author, too
- Examples:
  - Prevent channel change (by application, not user)
  - Prevent display usage (which could obscure video)

Announcement

- Advance notification of service information
- Used to make EPG and scheduling decisions by both receiver and viewer
- Placed in EIT and optionally, DET
- Compatibility Descriptor primarily, but also
  – Title, start time, and duration, if they are unique

Signaling

- Real time information about the transport resources
- Includes
  - appID
  - Compatibility Descriptor
  - Identifiers
  - Content Types
  - Broadcaster Permissions

Encapsulations

- (Asynchronous only for now)
- Asynchronous non-flow controlled scenario of the DSM-CC Download protocol encapsulated in DSM-CC sections
- Non-streaming Synchronized Download protocol encapsulated in DSM-CC sections
- Asynchronous IP datagrams in Addressable Sections
- Proprietary Data Piping

Summary

- Application Reference Model
- Glues A/90 with DASE
- Extends/Constrains A/90
- Provides Data and State Models
- Provides uniform resource naming
- Needed for interoperable implementations

Author Information

Michael A Dolan

miked@tbt.com

T3/S18 Chair

Thanks to DIRECTV for support in the general field and work with ATSC in particular.

DASE Security

Taylor Kidd

OpenTV
tkidd@opentv.com

As embedded processors and digital communications come to dominate the world that surrounds us, computer and digital security plays an increasingly important role. Today, television is transitioning into this digital world as various private and public organizations throughout the world struggle to define and implement the infrastructure needed to bring digital TV to every household. Along with the many remarkable advantages of using digital information (e.g. almost zero information loss, increased noise tolerance, accompanying programs), there are also risks due to the complexity and remarkable malleability of digital data. As such these drafting organizations, including the A TSC (Advanced Television Systems Committee) T3/S 17 Specialist Group -sometimes referred to as the DASE Specialist Group -are working to include elements of digital security in their specifications.

This presentation briefly outlines and introduces digital security, covering threats, services and mechanisms. After discussing the security approach of some of the different DTV (Digital TV) specifications being developed around the world, it focuses in on the security approach of the DASE (Digital TV Application Software Environment) Level 1 draft specification. Subsequently, the presentation concludes with some of the scenarios and approaches under consideration in DASE Level 2 security.

DASE Security

Taylor Kidd (OpenTV)
tkidd@opentv.com

19 June 2001

Outline

- Objective of Security
- DASE L1 Security
- DASE L1+
- Architecture

Objective of Security

Protect and Assure

- – Confidentiality
- – Integrity
- – Availability

Against Threats

- – Competitors
- – Hackers
- – Thieves
- – Etc.

Security Services

- Peer Entity Authentication
- Access Control
- Data Integrity
- Data Confidentiality
- Non-repudiation

Peer Entity Authentication

Determine identity of participants
- Unilateral Authentication
- Mutual Authentication

E.g.
- Bad Guy replaces his own bogus “Instant Winner” answer file for the Gilligan Game Show before the signal is broadcast.
- B. Guy substitutes in a bogus update file containing 1-800 phone numbers for USA TV-banking before they’re broadcast.
- B. Guy sends a fake transaction to Jake’s bank via the return channel.

Access Control

Regulate User Access to Resources

- Used with other services
- Log user activities

E.g.

- A Bad© Cola ad application attempts to access proprietary statistical data saved by Coke ads.
- A Bad© Cola ad application attempts to access proprietary code and data associated with a Pepsi Cola ad.
- A Bad© Store ad application attempts to access Jake's home email address saved on the receiver to sell to spammers.

Data Integrity

Insures Data Quality

- Practically means data sent is data received
- Not Peer Entity Authentication

E.g.

- B. Guy modifies the "Instant Winner" answers for the Gilligan Game Show before the signal is broadcast.
- B. Guy modifies the file containing 1-800 phone numbers for USA TV-banking before they're broadcast.
- B. Guy modifies a transfer transaction to Jake's bank sent via the return channel.

Data Confidentiality

Protects against unauthorized access to information

E.g.

- – B. Guy views a pay-per-view Boxing without paying.
- – B. Guy views the “Instant Winner” answers broadcast for the Gilligan Game Show.
- – B. Guy monitors Jake’s credit card number during a TV-Purchases, Inc., transaction across the return channel.

Non-repudiation

Protection from denial

E.g.

– After an e-commerce transaction with TV-Purchases, Inc., B. Guy denies the transaction took place.

DASE L1 Services

Has

- Access Control
  - Limited usefulness because there is no peer entity authentication

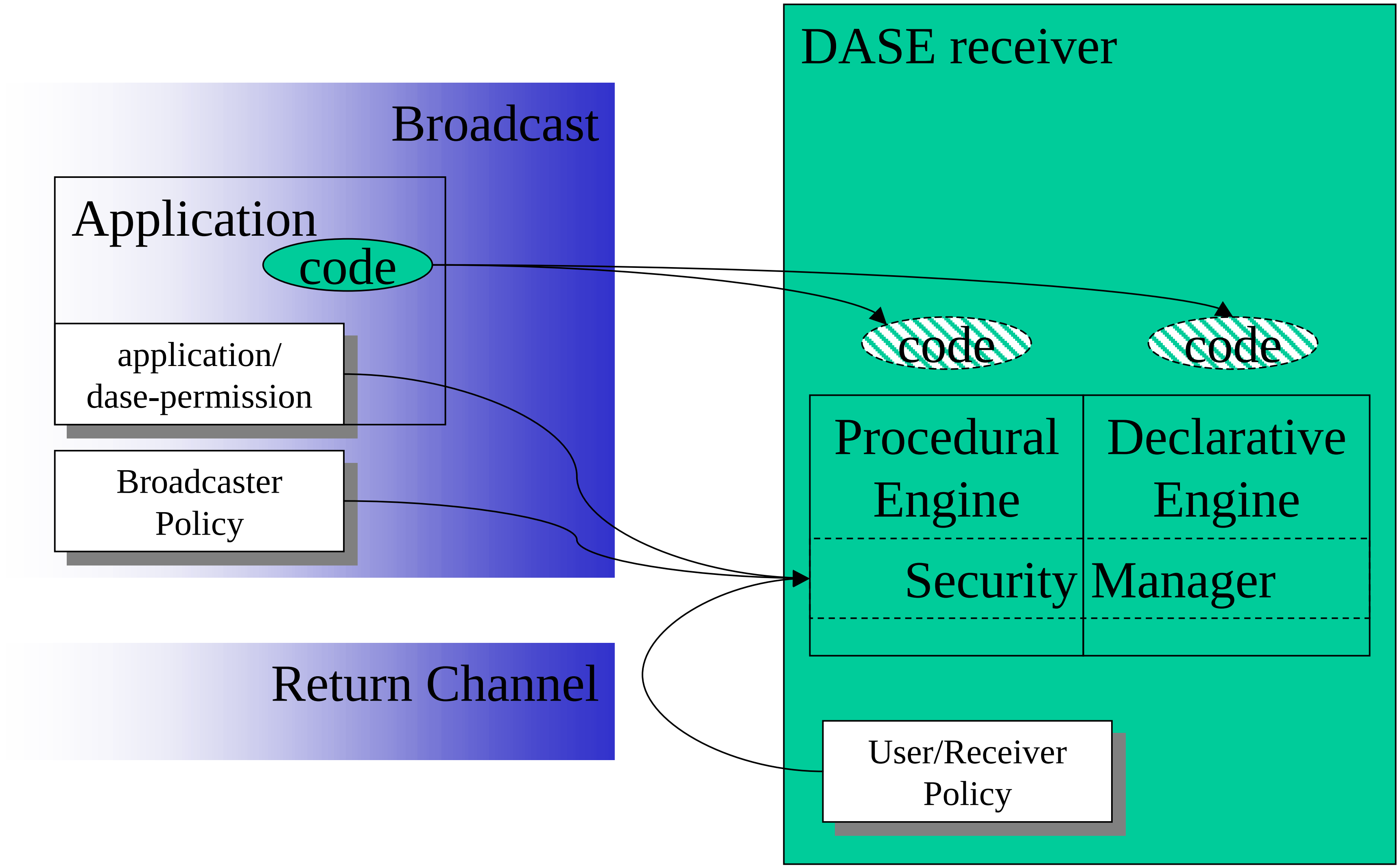
Doesn't have

- Peer Entity Authentication
- Data Integrity
- Non-repudiation
  - not relevant w/o return channel
- Data Confidentiality

DASE L2 Services (potentially)

- Peer Entity Authentication
- Data Integrity
- Access Control
- Probably handled in part or in whole by outside entities
  - Data Confidentiality (e.g. encryption)
  - Non-repudiation (e.g. transaction record)

Broadcast Environment



Security Architecture

- Procedural Engine
  - Code in Java
  - Uses Java 2 Security Model
    - Security Manager
    - Policy Object
    - Permission Objects
  - Well defined and tested
- Declarative Engine
  - Code in
    - XDML
    - ECMAScript
  - Policies
    - application/dase-permission policy
    - Broadcaster permission policy
    - Viewer/receiver policy
    - Legacy applications
  - Security
    - Implementation Dependent

application/dase-permission

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Permission PUBLIC "-//ATSC//DTD DASE Permission 1.0//EN" "">
<Permission>
  <Request Name="File" Target="/com/tv/info.dat" Actions="read,write"/>
  <Request Name="ServiceSelection" Target="*" Actions="*"/>
</Permission>
```

- XML Application
- List of “Request” objects identifying grants
- Bundled with TV program

application/dase-permission: DA

Operation or Feature	**Designation**	**Request Name**
Cookie Create, Delete, or Modify	grantable	Cookie
Event Attribute Create or Modify	grantable	RuntimeCodeExtension
Function.[[constructor]]	grantable	RuntimeCodeExtension
Global.eval	grantable	RuntimeCodeExtension
HTMLDocument.write	grantable	RuntimeCodeExtension
HTMLDocument.writeln	grantable	RuntimeCodeExtension
Script Element Create or Modify	grantable	RuntimeCodeExtension
Window.setTimeout	grantable	DelayedEvaluation
Xlet Instantiation	grantable	Xlet

application/dase-permission: PA

Permission Class	Designation	Request Name
java.awt.AWTPermission	denied	*none*
java.io.FilePermission	grantable	File
java.lang.ReflectPermission	denied	*none*
java.lang.RuntimePermission	denied	*none*
java.net.SocketPermission	grantable	Socket
java.security.AllPermission	denied	*none*
java.security.SecurityPermission	denied	*none*
java.util.PropertyPermission	grantable	Property
javax.tv.media.MediaSelectPermission	grantable	MediaSelect
javax.tv.service.ReadPermission	grantable	ServiceInfoAcces
javax.tv.service.selection.SelectContextPermission	grantable	SelectContext
javax.tv.service.selection.SelectPermission	grantable	Select
org.atsc.application.ApplicationPermission	grantable	Xlet
org.atsc.management.ManagementPermission	grantable	StateManagemen
org.atsc.preferences.PreferencePermission	grantable	Preference
org.atsc.registry.RegistryPermission	grantable	Registry
org.atsc.security.AtscAllPermission	denied	*none*
org.atsc.security.HAViPermission	denied	*none*
org.atsc.user.UserPermission	grantable	User

End of Presentation

NIST DASE Development Environment

Robert Snelick

Information Technology Laboratory
National Institute of Standards and Technology
rsnelick @nist.gov

The NIST DASE Development Environment is a collaboration effort of the National Institute of Standards and Technology (NIST) and the Advanced Television Systems Committee (ATSC) T3/S17 industry consortium for the proposed Digital TV Applications Software Environment (DASE) standard. NIST is directing their efforts towards the development of an ATSC Set-top Box simulation, a prototype implementation of the DASE Procedural Application Environment (PAE) Application Programming Interfaces (APIs) and reference applications. The intended use of the development environment is to demonstrate proof of concept of the DASE standard, provide the impetus for conformance testing, aid the design and development of other DASE implementations, and provide an environment for developing and testing DASE content/applications. In alignment with these goals, the design of the development environment emphasizes implementation clarity and portability over performance and system constraints. To achieve these goals, the majority of the system is written in Java. The NIST DASE Development Environment includes a runtime interface so that DASE Xlets can be easily created, run, and tested. All NIST produced source code, documents, and associated tools are placed in the public domain.

The core component of the development environment is an implementation of the DASE P AE. NIST has implemented the javax.tv, org.atsc, org.havi, and org.davic APIs. The PAE API implementation is currently built on top of the NIST STB simulation. The simulation is a collection of Java classes that encapsulate the functions of an ATSC STB environment. A central task of the Java simulation classes is to provide the implementation with ATSC data structures and associated data managers. A key aspect of the API PAE implementation design is an intermediate software layer, called the Hardware Abstract layer (HAL). The HAL provides an interface to the STB environment that hides the details of the underlying architecture from the implementation. It is envisioned that this multi-layered design will ease the task of porting the implementation to other receiver platforms.

The NIST DASE Development Environment also includes example native DASE applications, Xlets, and developer tools. Native applications include implementations of an Electronic Program Guide (EPG) and Channel Browser. Example Xlets include a Stock Ticker, E-Commerce, and a Service Provider EPG with in-band tuning capabilities. The developer tools include a stream injector, PSIP browser, and an Xlet controller.

NIST DASE Development Environment

Robert Snelick

DASE 2001 Symposium

NIST: The Who and the What

- Department of Commerce
- Information Technology Laboratory
- Assist U.S. Industry
- Forward-looking Standards
- Research

National Institute of Standards and Technology

Outline

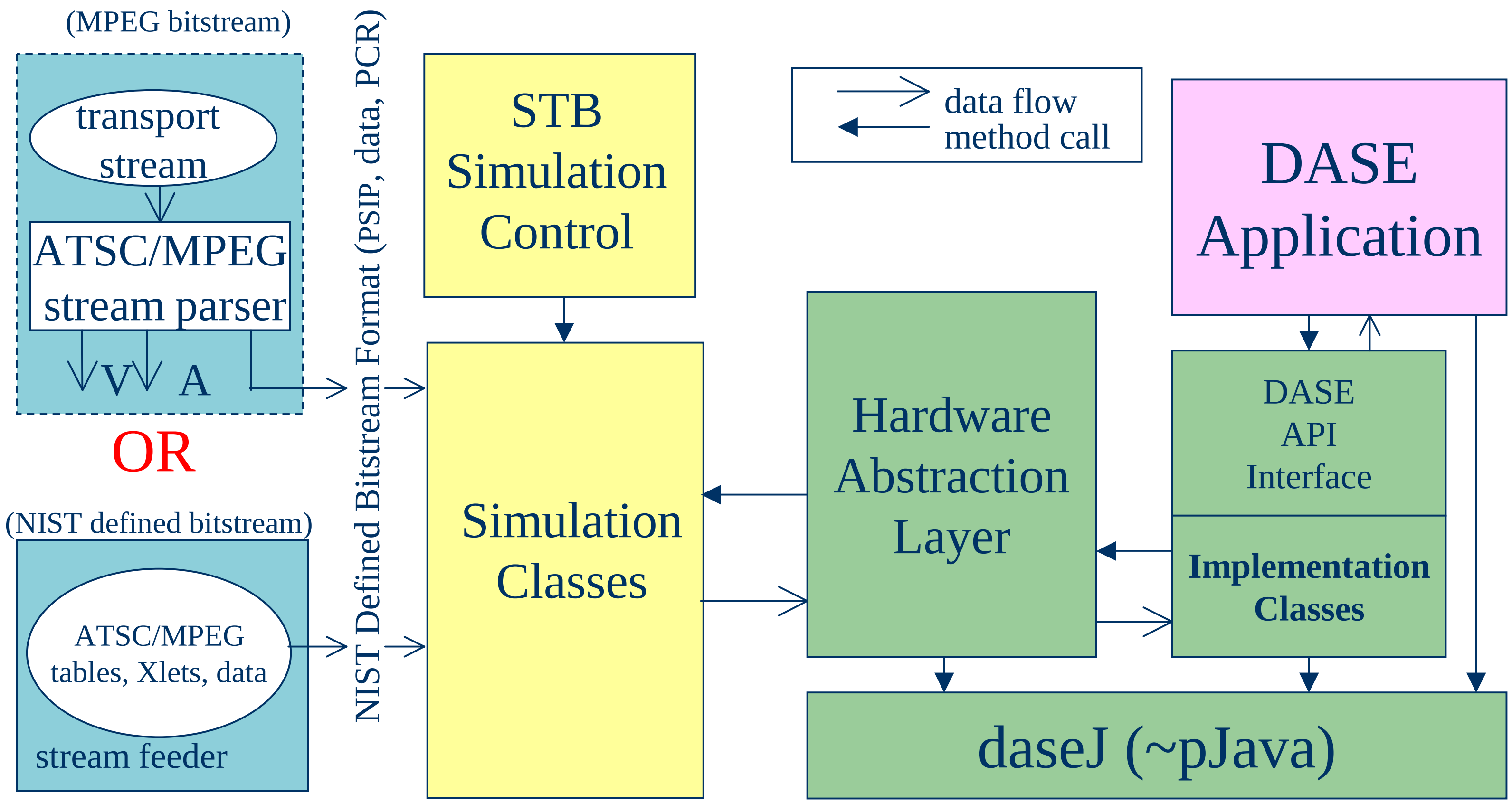
- Overview and Motivation
- Development Environment
  - STB Simulation Platform
  - PAE Prototype Implementation
  - DASE Native Applications and Xlets
  - Developer Tools
- Future Work
- Summary

National Institute of Standards and Technology

What is NIST doing?

- ATSC STB Simulation
- PAE Prototype Implementation
- Example DASE Native Applications and Xlets
- Developer Tools
- Bundled together as a Development Environment

NIST DASE Development Environment Architecture



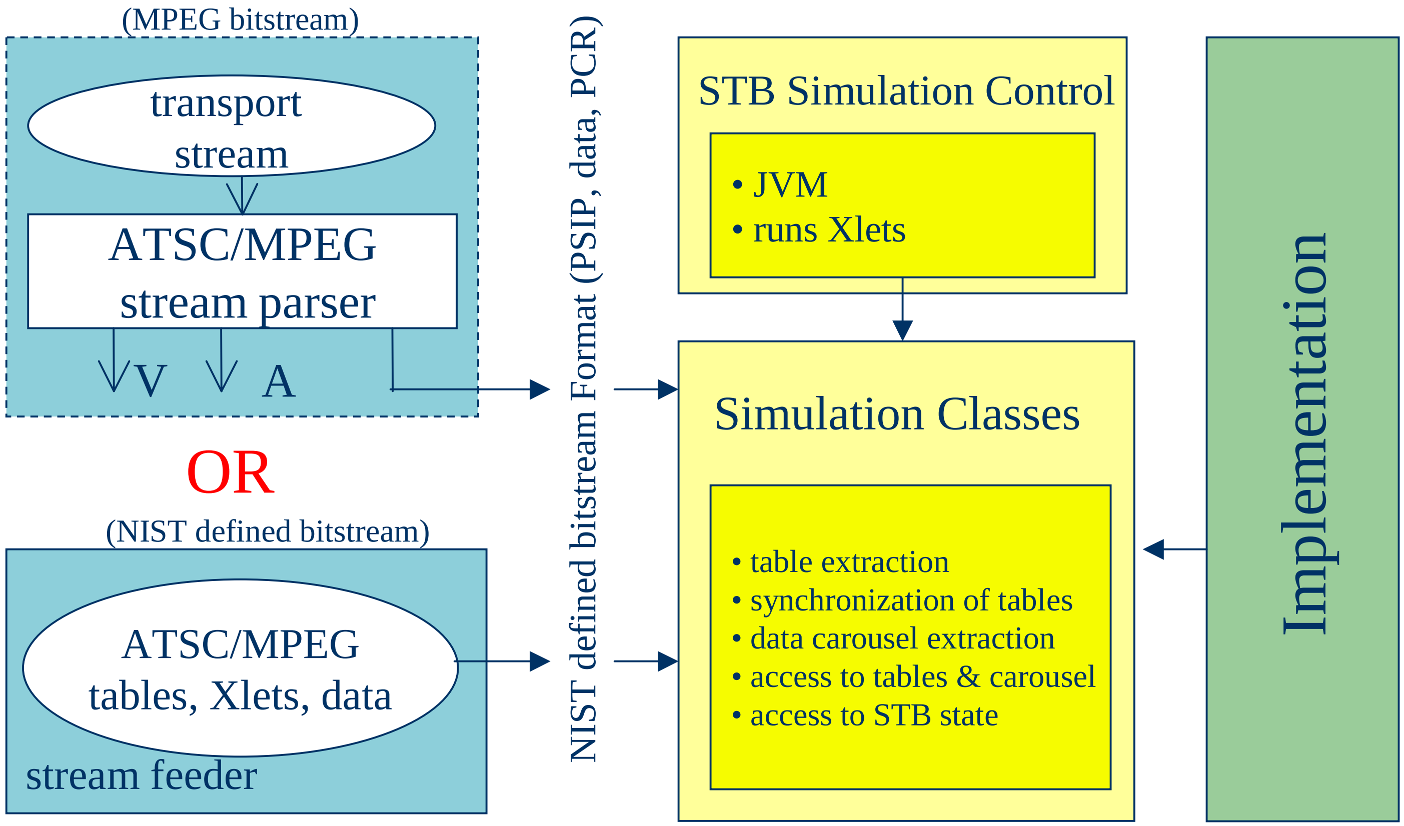
Benefits of NIST Environment

- Proof of Concept
- Conformance Testing
- Application Development and Testing
- Prototype Source Code
- NIST is Neutral, 3rd Party
- Public Domain

STB Simulation

- Java Simulation of an ATSC STB
- Independent of other system components
- Consumes streams containing ATSC/MPEG tables
- Data sink for API implementation
- Maintains table consistency
- Performs data management, not information management
- Extracts modules from the Data Carousel
- STB Simulation is NOT real-time

STB Simulation Components



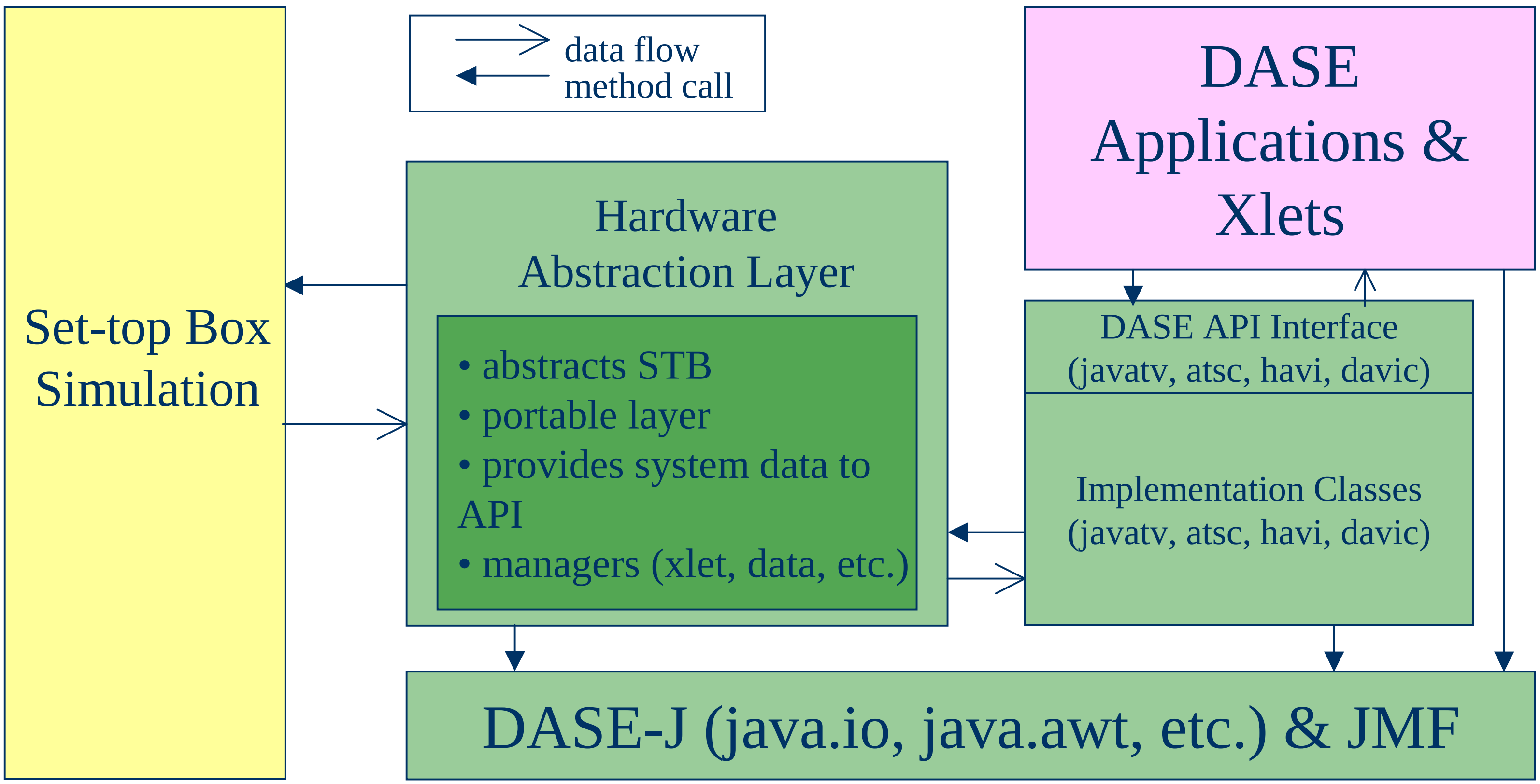
NIST
National Institute of Standards and Technology

PAE Prototype Implementation

- DASE-J (pJava 1.2, plus and minus)
- JMF 1.0
- Java TV (including JMF Player to STB)*
- ATSC *
- HAVI *
- DAVIC *

* Implemented by NIST

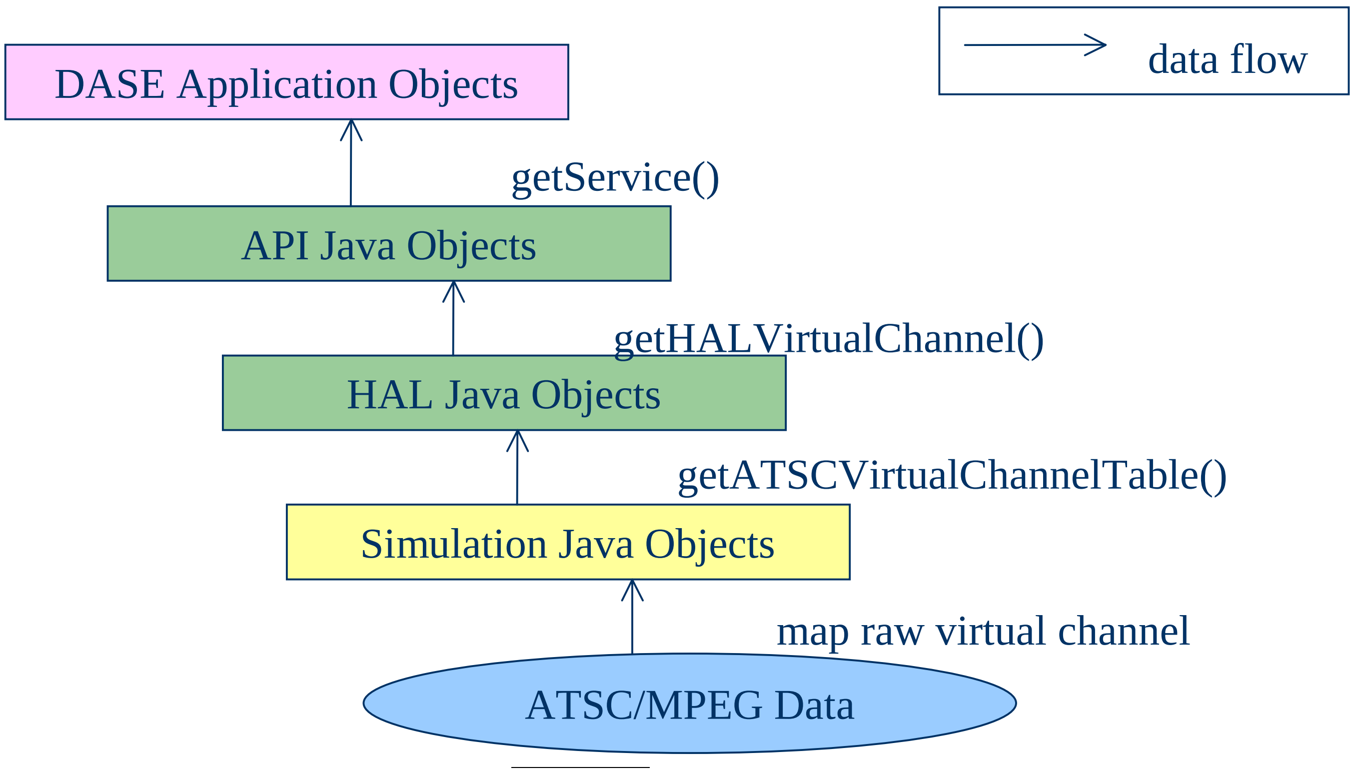
PAE Implementation Architecture



Hardware Abstraction Layer

- Intermediate software layer between API implementation and STB environment
- Common interface that abstracts lower layer
- Enables portability
- Transforms meta-data to API objects
  - Merges ATSC/MPEG tables
  - Maps to API objects

Data Flow Example



NIST HAVi Implementation

- Standalone implementation
- Uses java.awt light-weight components framework
- HAVi 1.0 currently
- Migrating to HAVi 1.1
- Framework complete with base set of widgets
- Fully compatible with AWT components

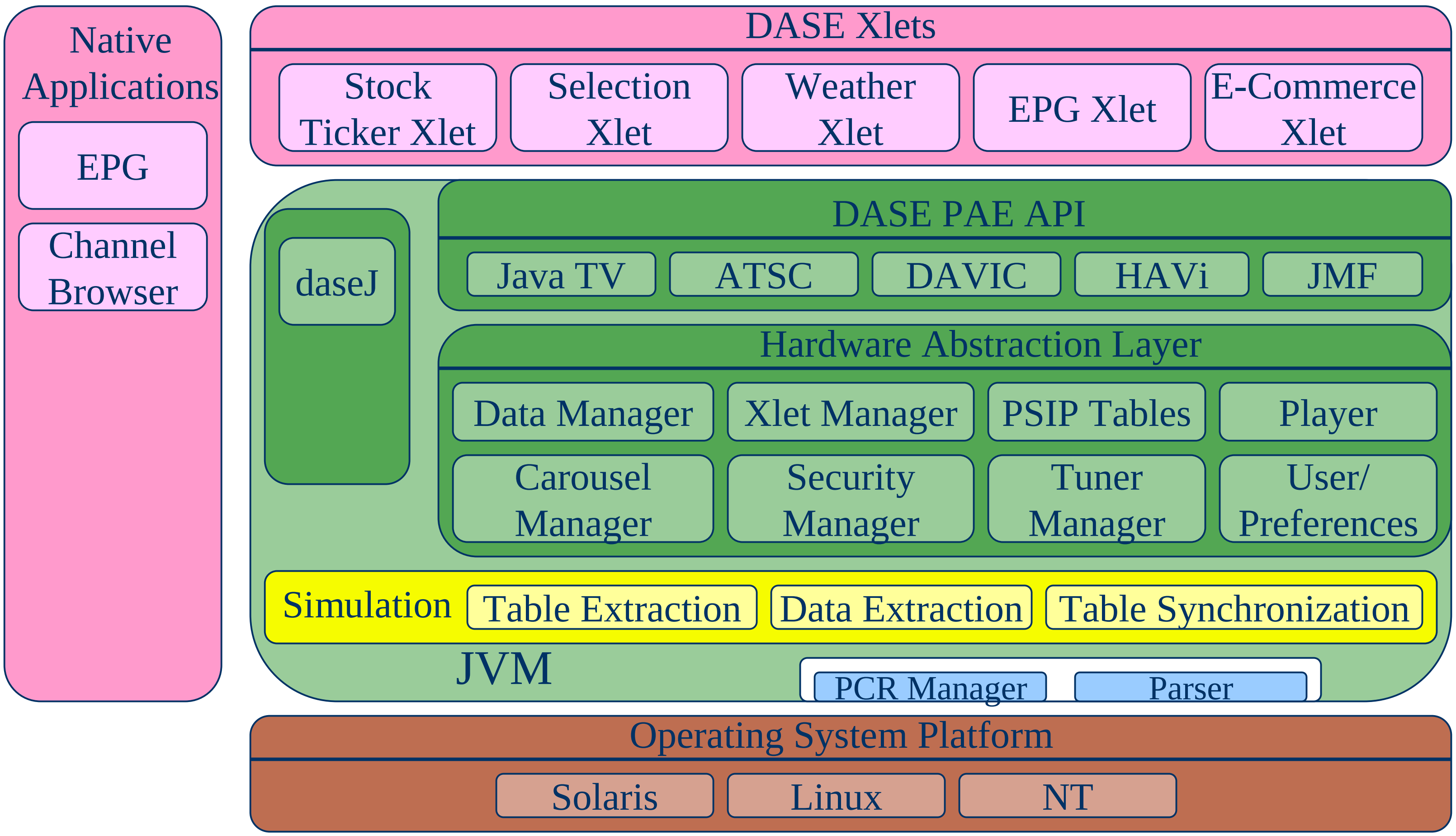
HAVi Implementation Framework

- Screen and Device Management
- Base Components and Containers
  - HComponent, HContainer, HVisible
- Simulated HAVi Compliant System
  - HScreen: simulated STB
    - Background device (still image)
    - Graphic device (including basic window management)
    - Video device (not implemented)
- All Standard Mattes
  - Flat and Image
  - Still and Animated

DAE "Implementation"

- Basic support for file carousel content
- DAE Application Manager
- Extracts modules from Data Carousel
- DAE Framework (interface for browser)
- Displays content (currently supports HTML)
- Implemented with Java Swing (renders basic HTML)

NIST Environment Summary



Implementation Status

- STB Simulation
  - functions necessary for implementation
- PAE Implementation
  - prototype implementation
  - missing/evolving functionality (security, ARM, etc.)
  - conformance tests forthcoming
- DAE Prototype Implementation
  - framework
  - thin

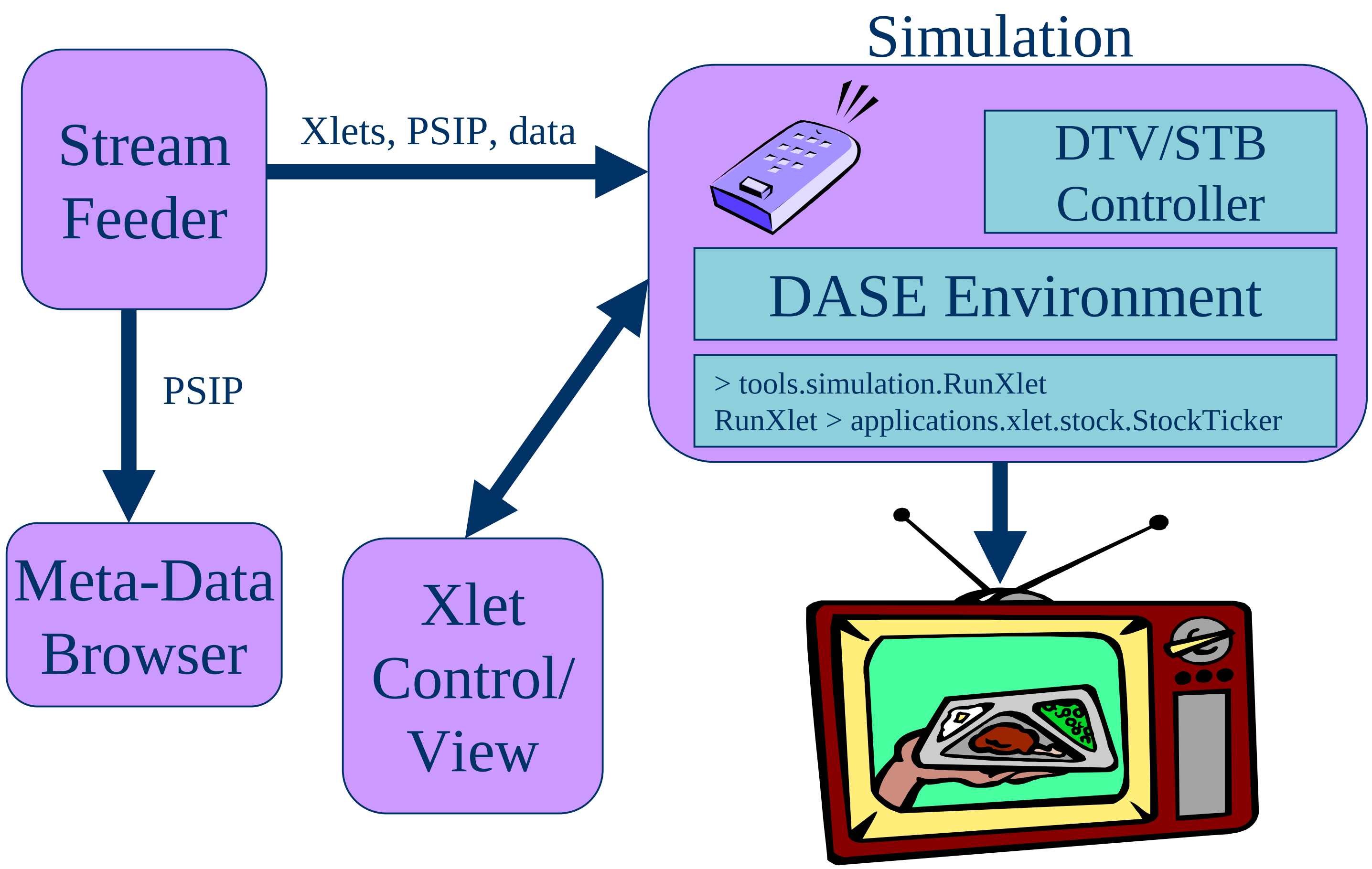
DASE Applications and Xlets

- Electronic Program Guide (EPG)
- Channel Browser
- User Preferences
- Stock Ticker Xlet
- Provider EPG Xlet (tuning via the API)
- E-Learning Xlet
- E-Commerce Xlet (HTML, no back-channel)
- Weather Xlet

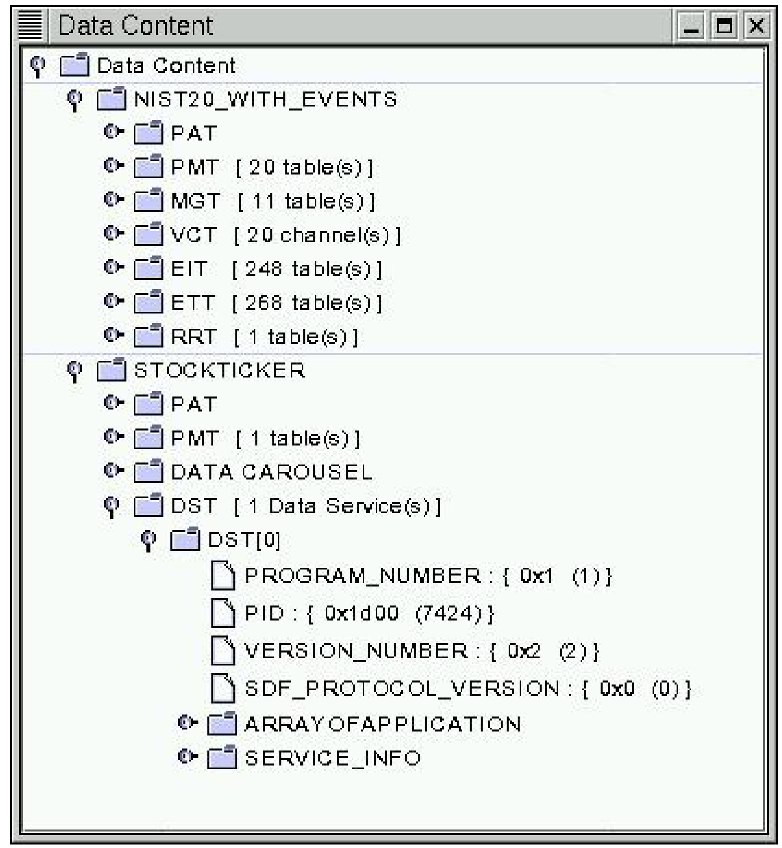
Developer Tools

- Stream Encoder (Simulation)
- Transport Stream Feeder
- Meta-Data Browser
- Software MPEG Parser (De-multiplexor)
- API Unit Tests
- DTV/STB Simulator (remote/controller/display)
- RunXlet
- Xlet Manager Viewer/Controller

Runtime Environment



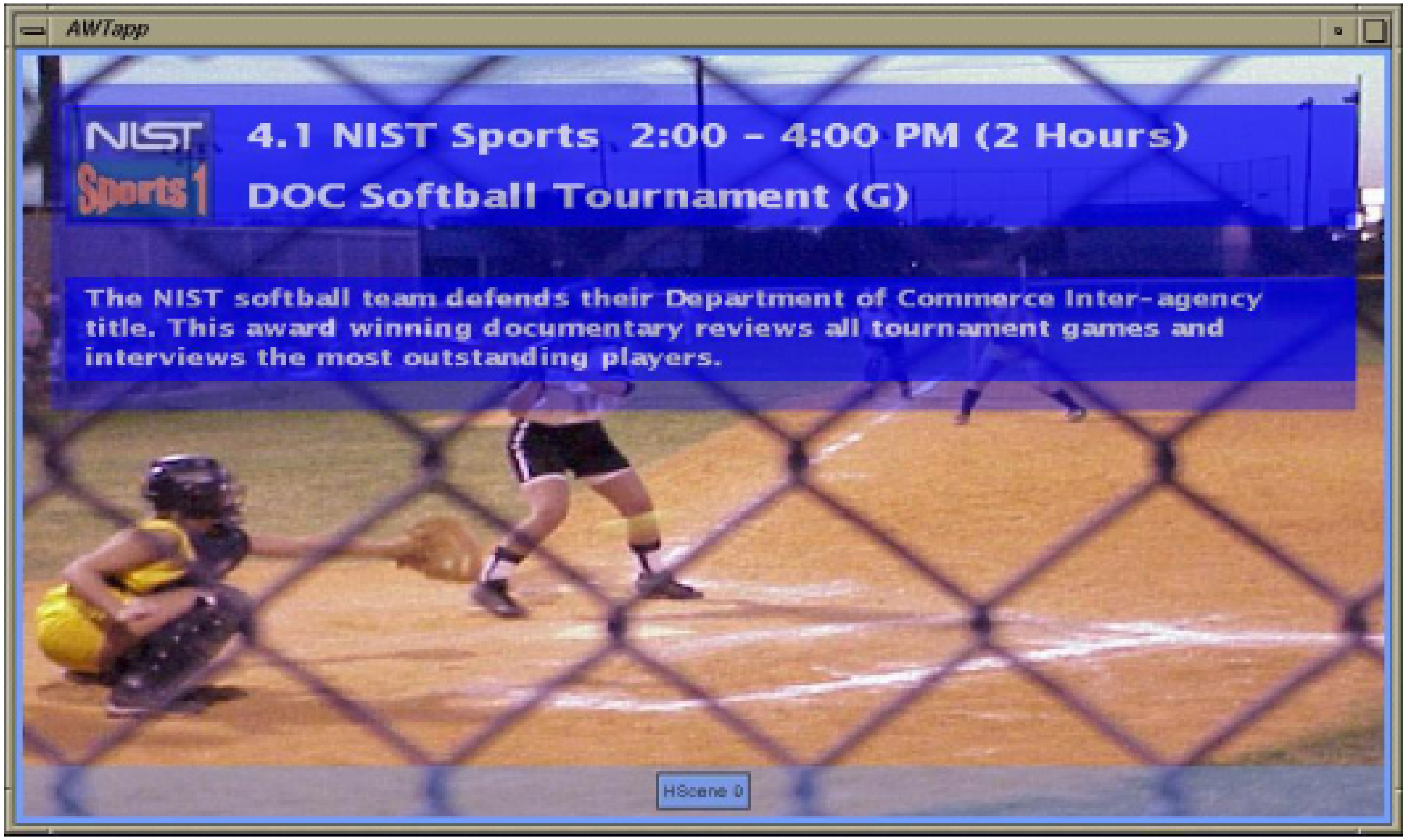
Stream Feeder/Meta-Data Browser



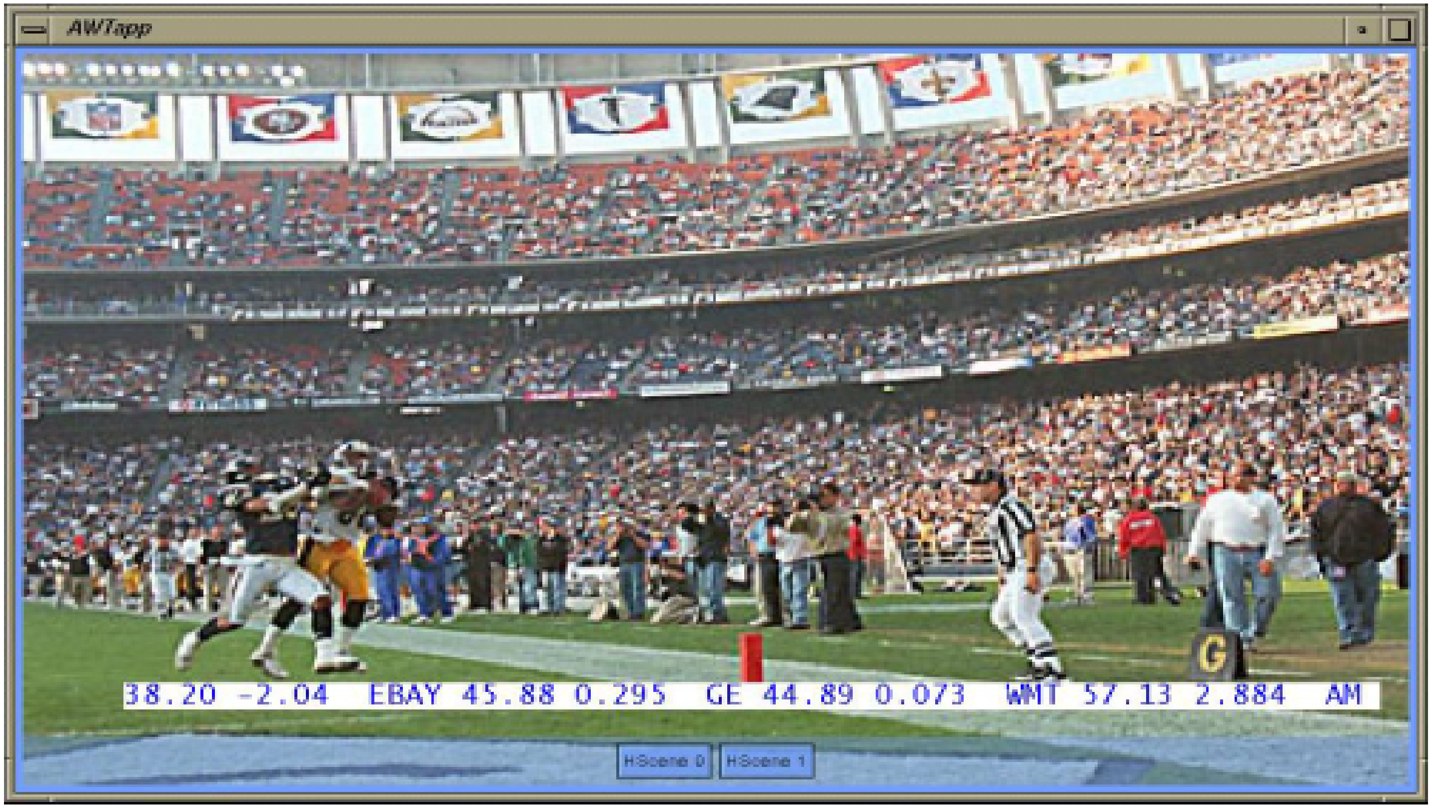
NIST
National Institute of Standards and Technology

Electronic Program Guide (EPG)

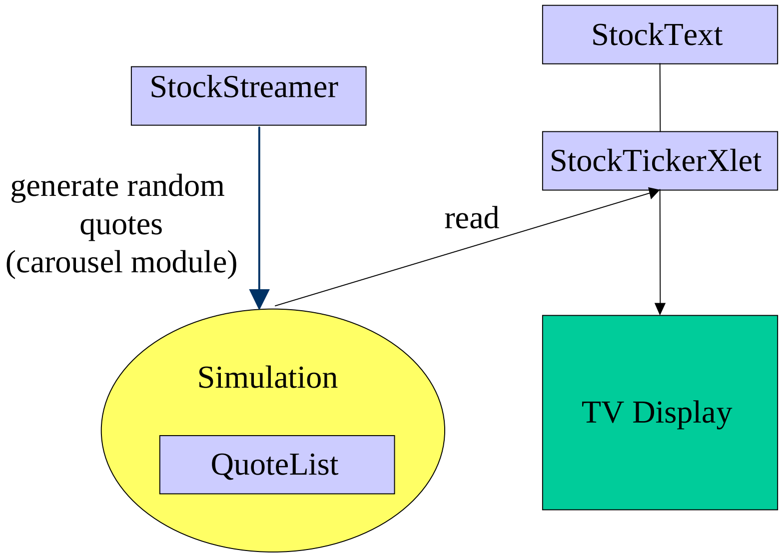
Channel Browser



Stock Ticker Xlet

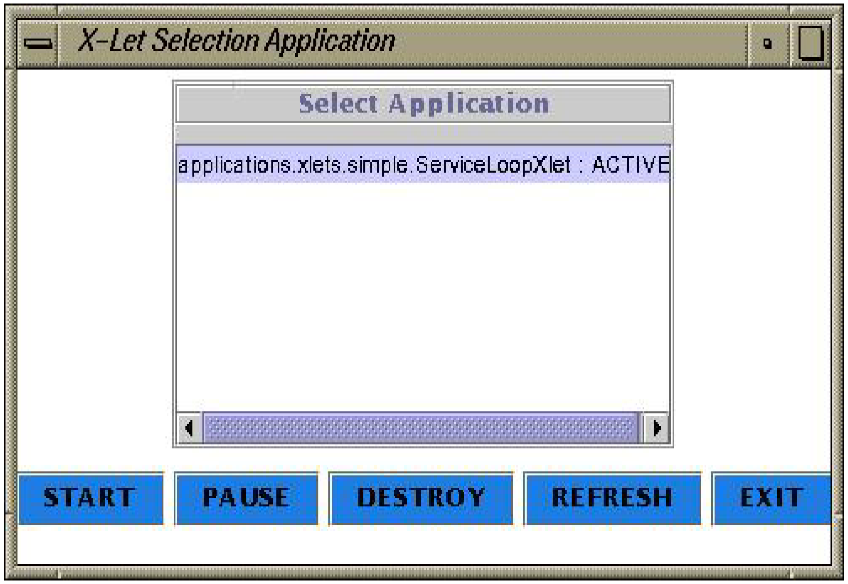


Stock Ticker Xlet



- Implement Xlet interface
- Acquire HAVi HScene
- Open Carousel File
- Build HAVI scrolling text
- Start Xlet
- Add Carousel Listener
- Refresh Cache
- Read Quotes
- Update scrolling text

Xlet Viewer/Controller



- Debugging tool
- Indicates the status of Xlets
- Start, pause, and destroy Xlets
- Handles multiple Xlets

National Institute of Standards and Technology

Resources

- *http://www.dase.nist.gov*
- Implementation Source Code
- Data Sets
- User's Guide
- NIST Implementation Guide
- Java Doc
- Xlet descriptions and source code

NIST
National Institute of Standards and Technology

Future Work

- Tie-up Loose Ends
- Port to real-time STB
- Performance Measurements
- Develop Metrics
- DAE Implementation
- DASE-2

National Institute of Standards and Technology

Summary

- DASE Development Environment
  - STB Simulation Platform
  - PAE Prototype Implementation
  - Sample Xlets
  - Developer Tools
- Runtime Environment
- Prototype Source Code
- Application Development and Testing Platform

Team Members

- Alan Mink
- Robert Snelick
- Wayne Salamon
- Mike Indovina
- Michel Courson
- Guillaume Lathoud
- Guillaume Marcais
- Gaetan Guttermann

An Automated Approach for DASE Conformance Testing

Andrew Twigger

UniSoft Corporation
Millbrae, CA
andrew.twigger@unisoft.com

The successful implementation of digital television standards depends on the ability to consistently produce, broadcast and present content to the viewer. In an open market where multiple content providers, broadcasters and consumer electronic manufacturers are involved in the end-to-end process, the need for testing tools is paramount. The initial need to show component consistency and conformance to standards is essential to the successful development and implementation of digital broadcast systems. Once an installed receiver base is established and new equipment providing more advanced capabilities appears on the market. conformance-testing tools will be needed to ensure that end-to-end broadcast compatibility is maintained.

This presentation describes an Automated Test Environment that has been developed by UniSoft as a first step towards addressing the industry's need for testing tools. The Automated Test Environment provides a test laboratory emulation of a broadcast to the receiver and allows a test to simulate user interaction with the receiver through a remote control. The architecture of the Automated Test Environment is built using a flexible structure that is extensible for use in both receiver and application content testing. This flexible design provides clearly delineated component boundaries allowing the technology to be used in terrestrial, satellite and cable operations and for testing applications and system components written to conform to different API standards.

The current version of the Automated Test Environment is being used by five major consumer electronic suppliers to test their DVB MHP implementations. Extensions are already planned to cater for DASE and other digital television standards.

An Automated Approach for DASE Conformance Testing

Andrew Twigger
UniSoft Corporation
andrew.twigger@unisoft.com

Objectives

- Design a flexible and comprehensive Test Manager
- Implement a standard means to exchange information between the receiver and a computer workstation
- Provide support for a wide variety of test purposes
- Develop a security infrastructure suitable for use in a test laboratory

Terminology

- Assertion
- Test Suite
- Test Set
- Test Purpose [TP]
- Test Manifest
- TETware components
    - Test Case Controller [tcc]
    - Test Case Manager

Features and facilities

- Support for POSIX-style assertion-based testing
- Test scenarios can be defined using a powerful scenario language
- Test parameters can be specified using a flexible configuration variable mechanism
- Configuration information and test results are recorded in a journal
- Support for the standard POSIX results codes is built in

TETware Testing Model

- The test harness and the test purposes all run on the system under test
- The list of tests to run is read from a scenario file
- The results of tests are written to a journal

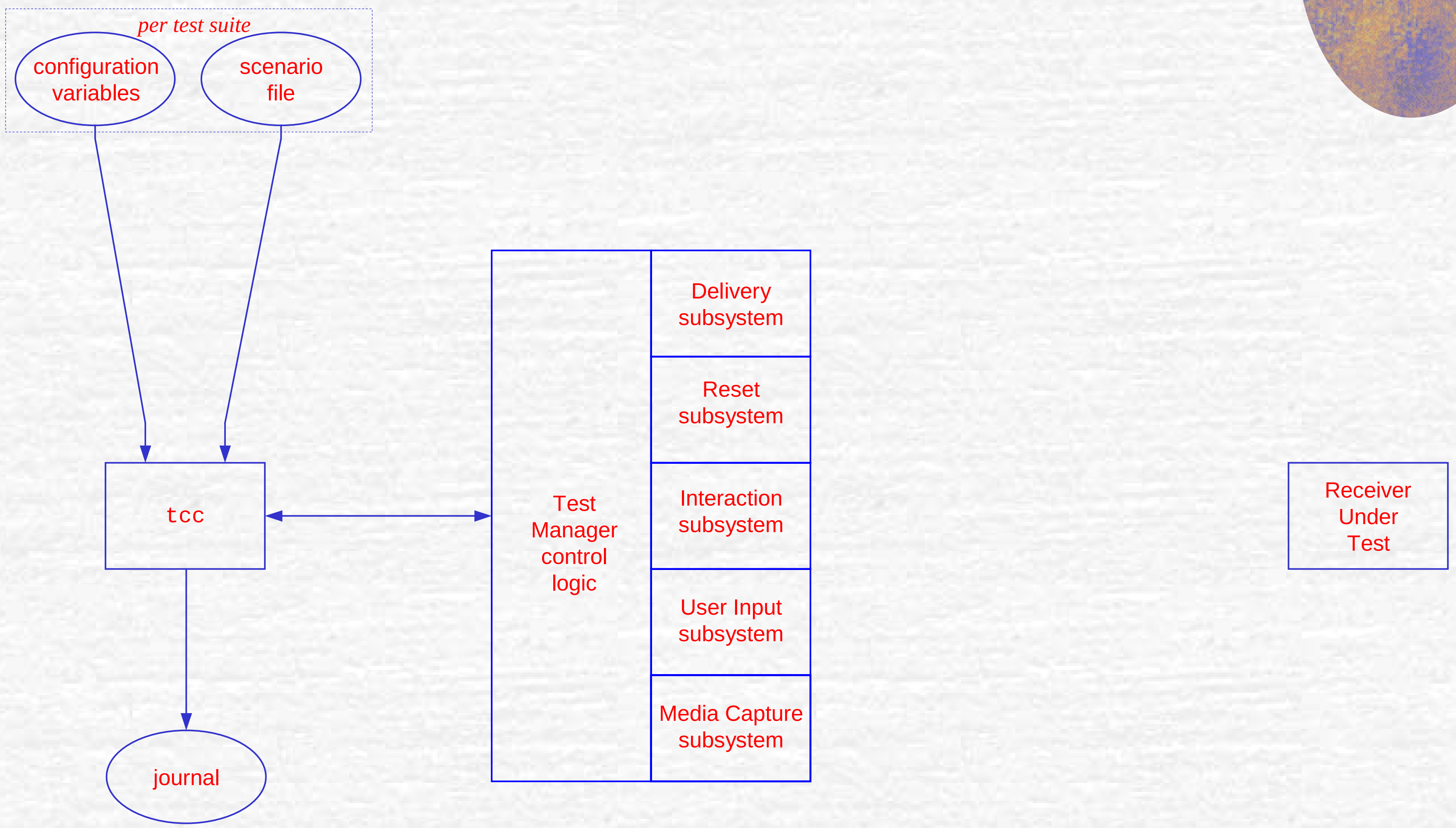
MHP Testing Model

- TCC does not run on the RUT, all control operations on a host system
- The test manager process provides the interface between the TCC, the STB, and the other hardware components

Test Manager

- Based on publicly available TETware 3.3
- Provides the interface between TETware and the receiver under test [RUT]
- Defined interface to hardware specific code

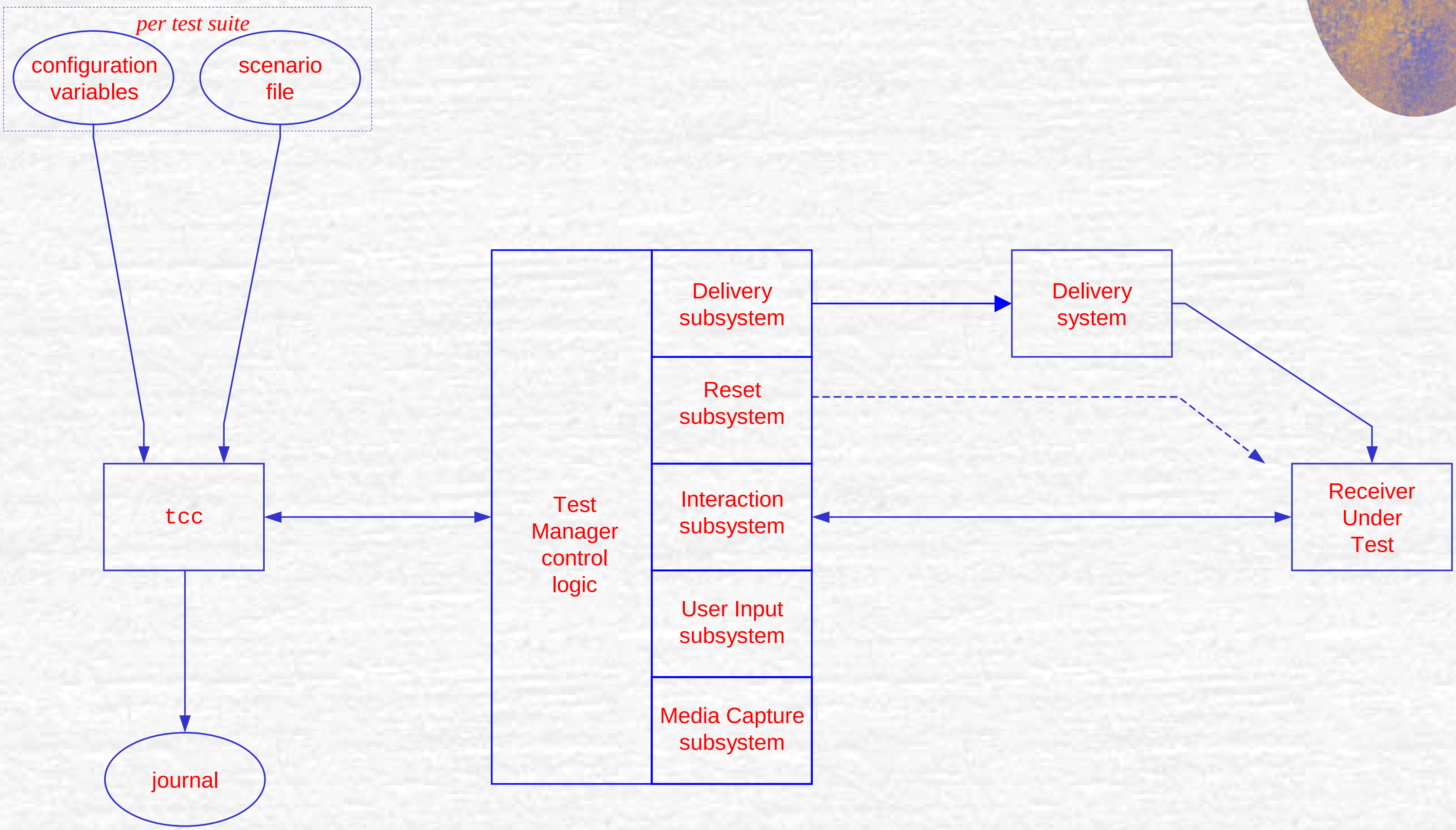
Test Manager block diagram



Test Manager operations

- Determine test set geometry
- Prepare broadcast streams
- For each TP function:
  - Reset the RUT
  - Play the initial broadcast stream
  - Enter a service loop
  - Write the result to a journal
  - Unload the broadcast stream
  - Perform reset operations

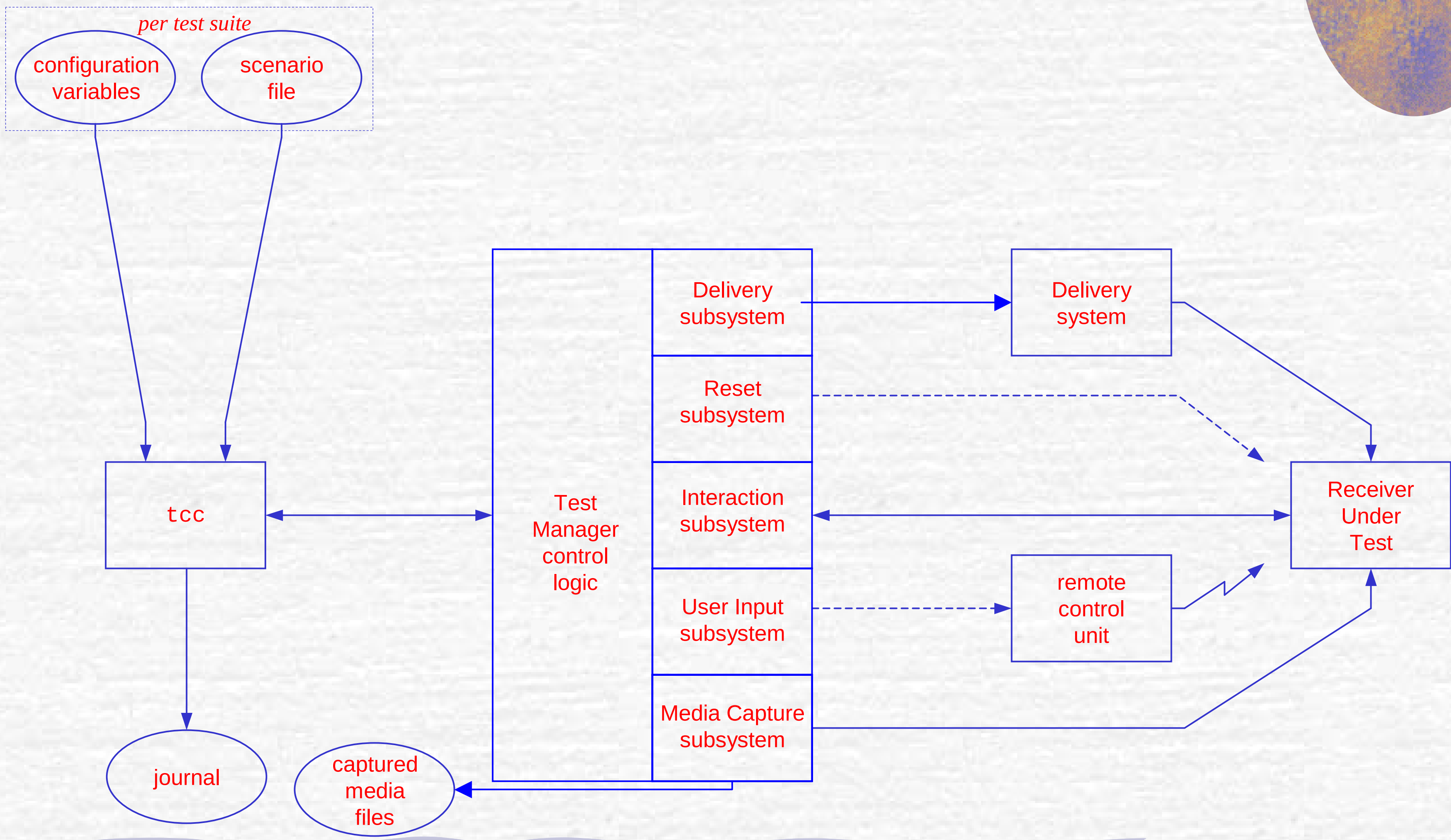
Test Manager block diagram



Test Manager Service Loop

- Respond to requests from the Interaction Subsystem
  - Start/Stop media capture operations
  - Press remote control button
  - Play out a transport stream
  - Log messages to the journal
  - Log result to the journal

Test Manager block diagram



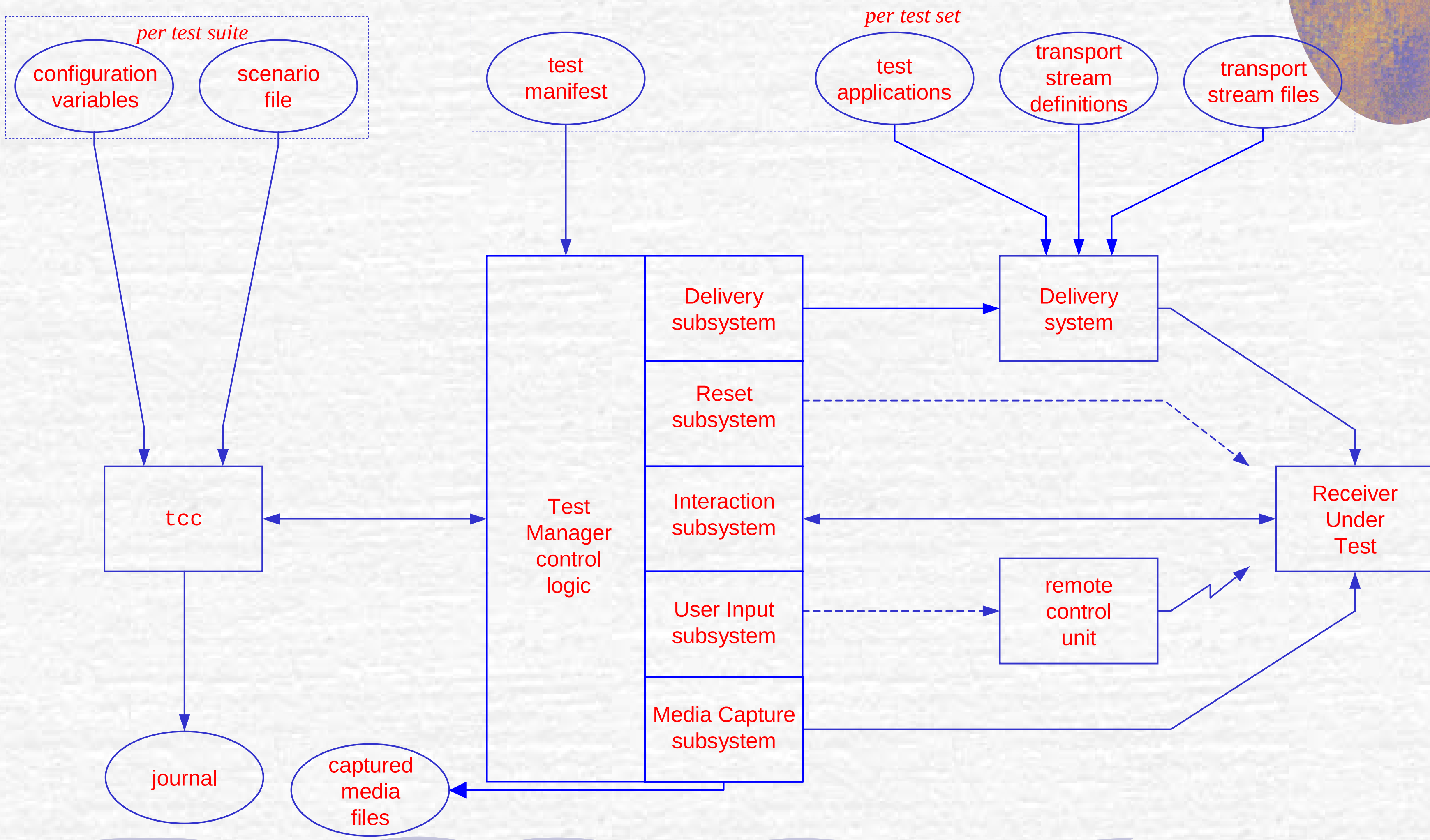
Test Set constituents

- Test manifest
- Each TP:
  - One or more transport stream description files
  - A java class – the “Testlet” class
  - Supporting java classes

Test Manifest

- Source of information about the test set
- XML document
- Describes:
  - **Test purpose numbers**
  - **Applicable MHP profiles and options**
  - **Configuration information affecting the execution of test purposes**
  - **Key words identifying each TP**
  - **Location of transport stream description files**
  - **Reset operations required by each TP**
  - **Test purpose time-out**

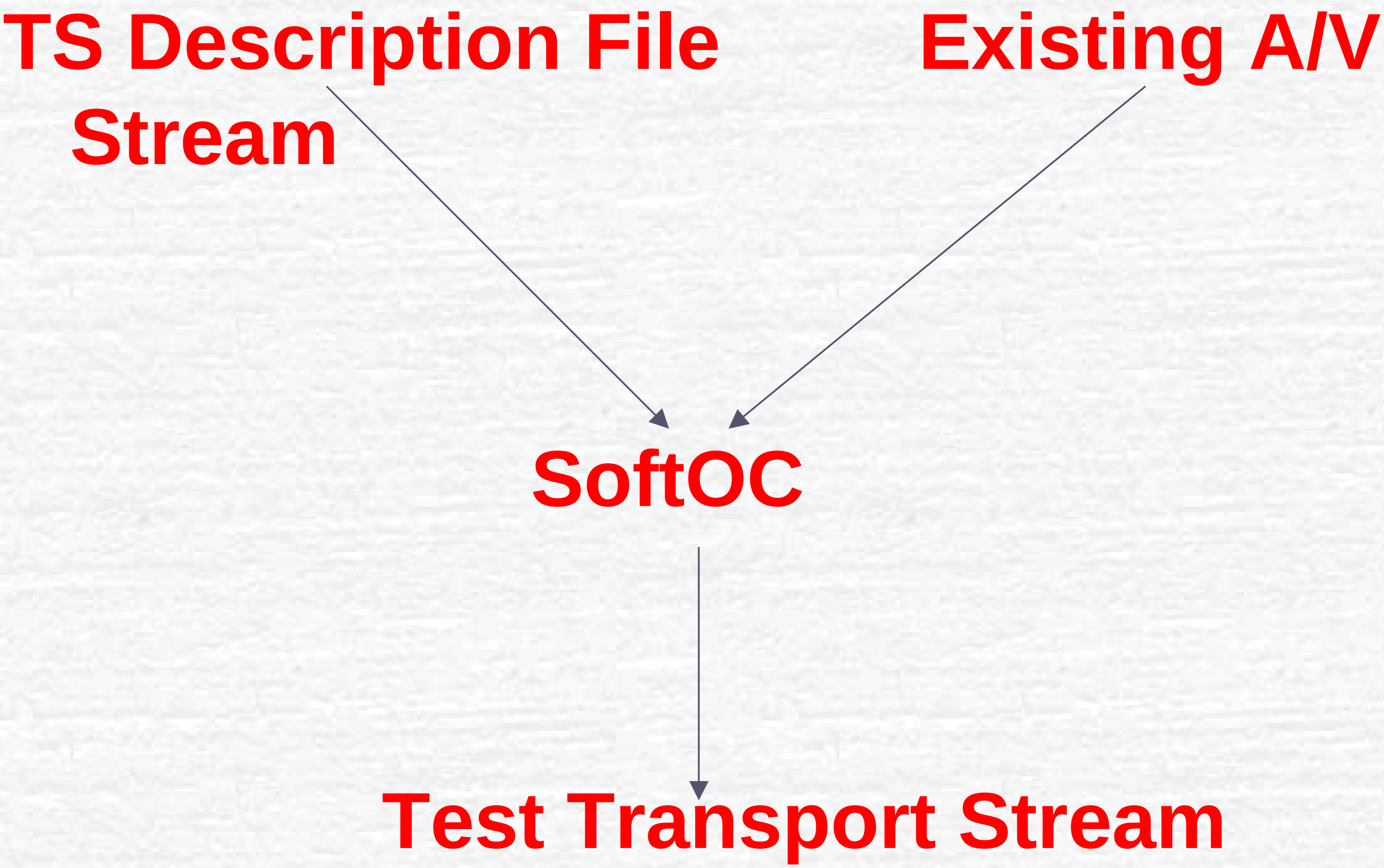
Test Manager block diagram



Transport Streams

- Each TP consists of one or more transport streams [TS]
- TSs identified by channel number and sequence number
- Typically short
- Designed to be played in a looping manner

Transport Stream Generation



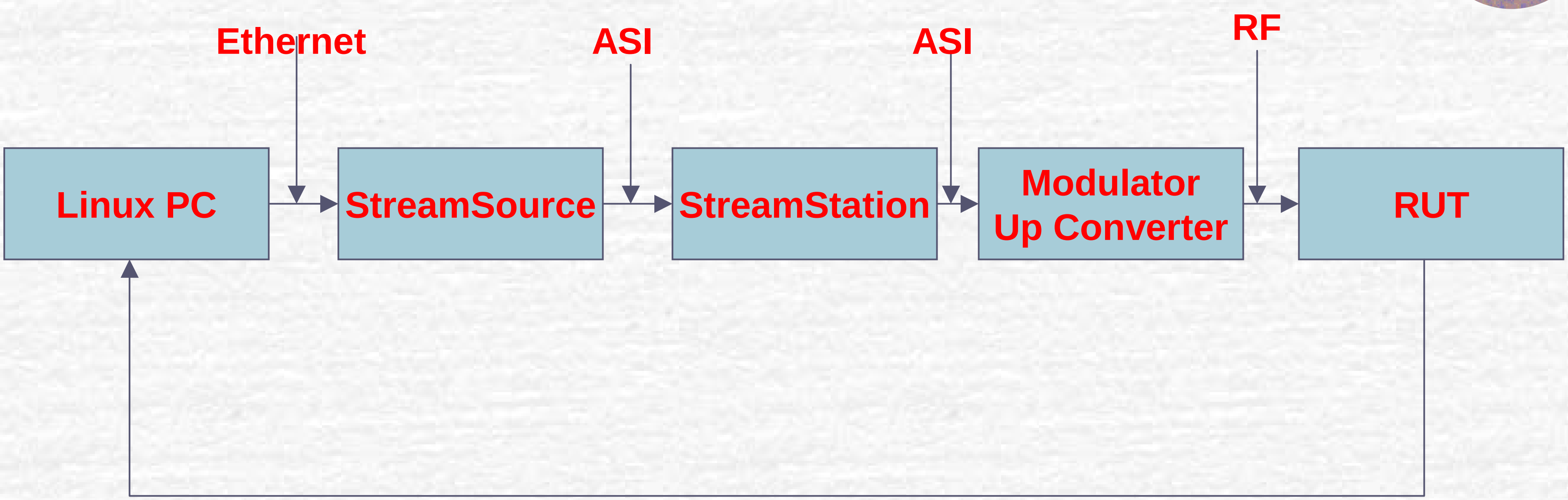
Transport Stream Generation

From SoftOC	From A/V Stream
PAT	MGT
PMT	EIT
VCT	STT
Application Information	Video
DSM-CC	Audio

Security Infrastructure

- Generate certificate chains (including badly formed certificates)
- Provide root certificates to receiver manufacturers
- Develop application signing tool and hash file generation tool
- Create CRLs and root certificate management files

Current Test Environment



PPP over serial connection

Planned ATE Extensions

- Automating the User Interaction
- Supporting Audio/Video Capture
- Return Channel support via a networked server
- Automating Test Manifest generation for application test capture and replay

ORBIT - OBJECT RECONFIGURABLE BROADCAST USING IT

Pedro Botelho Cardoso

INESC,Portugal,
pedro.cardoso@inescporto.pt, Tel: (+351) 222094200, Fax: (+351) 222084172

Recent years have brought digital to broadcasting. In such a conservative environment as television, changes are slow but consistent. After starting with transmission, archiving and playout, the move is now towards the post-production environment. Current processing power and network bandwidth envisage in the near future that broadcast solutions, based on proprietary technologies with limited multi-vendor integration, will be progressively replaced by new alternatives adopting an IT approach based on open architectures, low cost generic hardware, distributed and object-oriented paradigms.

INESC Porto and BBC investigated the suitability of a distributed architecture using an IT approach to broadcasting in the European project ATLANTIC (ACTS) and continued the work in ORBIT (a BBC funded project). ORBIT is intended to provide, over two years (1999/2001), in a pilot implementation, a small-scale model capable of handling "live" and recorded signals, from local and distant sources, of integrating media asset management and content handling tools and of demonstrating the facilities and the interconnections which will be needed in a full-scale operation.

THE ORBIT PROJECT

At the end of its first phase in April 2000, ORBIT had demonstrated the technical and economical viability of the proposed architecture. This work led to presentations and/or contributions in several organizations: SMPTE, MPEG and Pro-MPEG. An initial version of ORBIT is currently working at the BBC R&D laboratories and has attracted significant interest. As a result several new projects are looking at the ORBIT technology as a possible middleware solution. The demonstrator will be available to professionals to allow testing of the new methods and tools, for program production, while providing feedback for tuning of ORBIT.

THE ORBIT ARCHITECTURE

MBE (Multimedia Broadcast Environment), the core of the ORBIT architecture, is an object based middleware solution for the integration of essence and metadata in broadcasting environments. This framework provides the architecture for full-scale deployment of objects in a network environment independent of their nature by using CORBA and XML technologies. Where proven solutions and/or standards exist they are adopted into the ORBIT architecture. Some examples are: the ASCA (SMPTE) proposal to define the control architecture and the application programmer interfaces (APIs) for components comprising an advanced digital studio; the W3C tools for data representation (XML based)
and the MPEG-7 multimedia content description interface as the basis for data model development.

The post-production environment is the main ORBIT use case. The applications use Java Beans and ActiveX Components to demonstrate the functionalities needed: intake (live DVB and recorded material), logging, program manipulation and editing. Dual capture intake (low and high-resolution material) enables most of the operations to be carried with the low-resolution format reducing the required bandwidth. Final edit decisions can conform to high-resolution material using ATLANTIC techniques manipulating LongGOP MPEG2 compressed format.

CONCLUSION

Television post-production is possible today using IT technology. Post-production components can be implemented as software objects using CORBA and XML based middleware to establish control and communication between components and to integrate metadata and content handling. The key to interoperability however is the definition of the middleware interfaces and consistent metadata dictionaries and schema definition in international standards organizations.

ORBIT

(Object Re-configurable Broadcast Infrastructure Trial)

(Object Re-configurable Broadcast using IT)

Pedro Cardoso

INESC Porto

INESC PORTO

orbit

BBC R&D

A BBC perspective *

* Richard Storey (BBC R&D)

INESC PORTO

orbit

BBC R&D

Why the need to change?

- There's more competition than ever
- Programme budgets get tighter
- Viewer and listener expectations get higher
- Good programme content is the key *but:*
- Technology has a part to play

INESC PORTO

orbit

BBC R&D

How can technology help?

An efficient production system should:

- Provide information where and when it's needed
- Avoid *unproductive* delays
- Allow tasks to be done in the best order
- Make the fullest use of its resources

There is a *lot* of scope for improvement!

INESC PORTO

orbit

BBC R&D

So the changes that need to be made are:

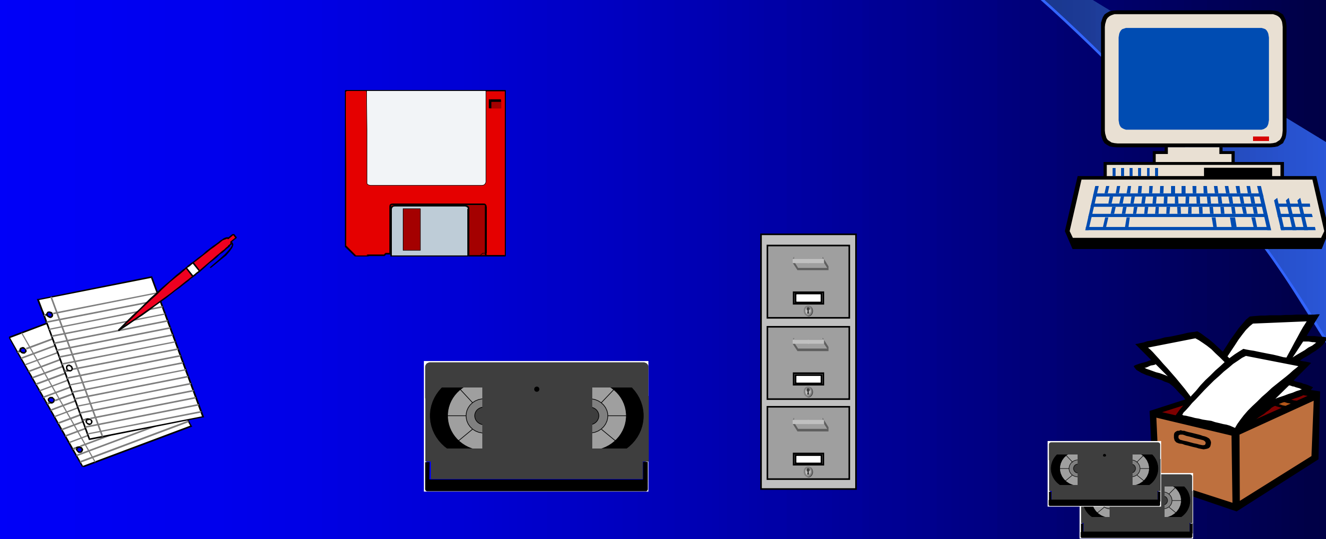
Re-engineer our *processes* to capture and retain the *metadata*, keeping it linked to the *media* so we can provide them both, wherever and whenever they are needed.

But the cost of doing this must be less than the amount of money that we can save, otherwise there's no *business case*.
(nobody in their right minds would pay for it).

INESC PORTO orbit BBC R&D

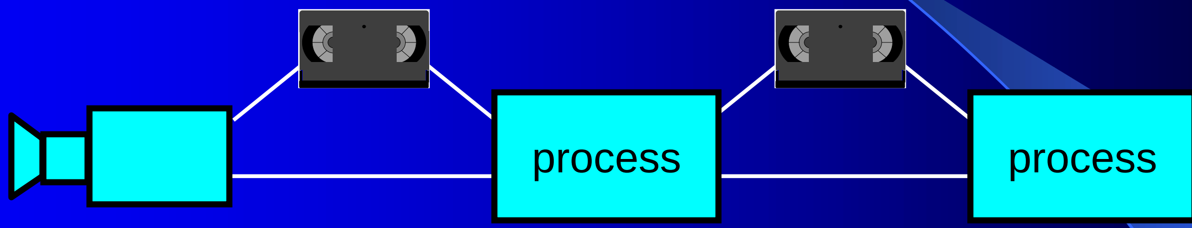
Before Process Re-engineering

Broadcasters have always had metadata - in many inaccessible and easy to lose forms; some of it has been entered and lost many times.



This is expensive, wasteful and demoralising.

But the media chain has worked well for decades.



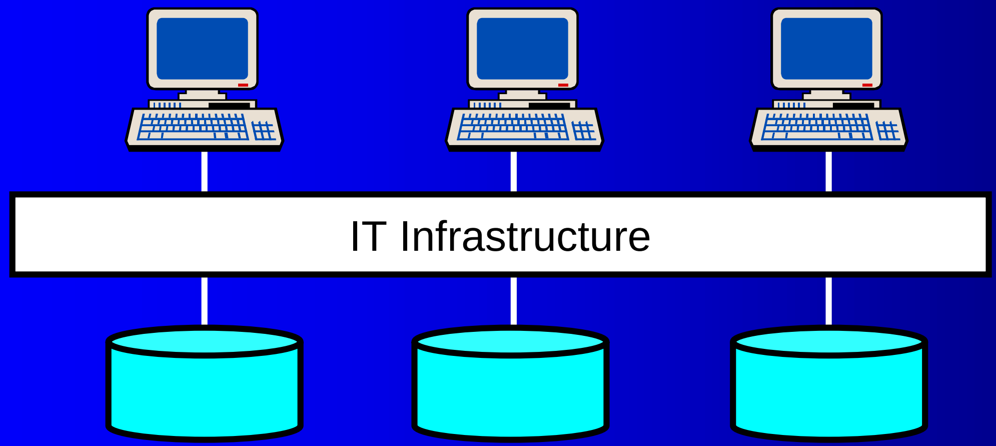
Because we have had standards:

Physical: VHS, C format, DI-5, Digi Beta, DVC……

Electronic: PAL, Rec. 656, SMPTE 270M, AES/EBU…….

Re-engineering the metadata is (relatively) easy.

- The amount of data is low,compared to the video
- Sounds like a job for IT



Distributed access to Metadata
(Data model entities)

Most tools are already there: SQL, HTML, XML...

INESC PORTO

orbit

BBC R&D

To Re-engineer the *metadata* you need:

Data Model to agree what the metadata means (SMEF - Standard Media Exchange Framework)

XML to pass it around the system, along with the definitions (W3C).

Databases and search tools to store it and find it again (various)

A way of linking it to the corresponding media - e.g. a UMID (SMPTE)

A way to implement your Business Rules, what data gets entered where, when and by whom (ORBIT)

A Project to put these all together and make them work (ORBIT)

INESC PORTO

orbit

BBC R&D

But what about the media?

Instant metadata is not enough. What if:

You've still got to wait a day for the tapes?

You've got to wait another day because the content was wrong?

Someone got the description wrong?

The tapes are out on loan, lost, mislaid, damaged, obsolete?

You've got to book a tape machine to view them?

You've got to view the whole tape to find the bit you want?

It takes three days to get your rushes onto the system?

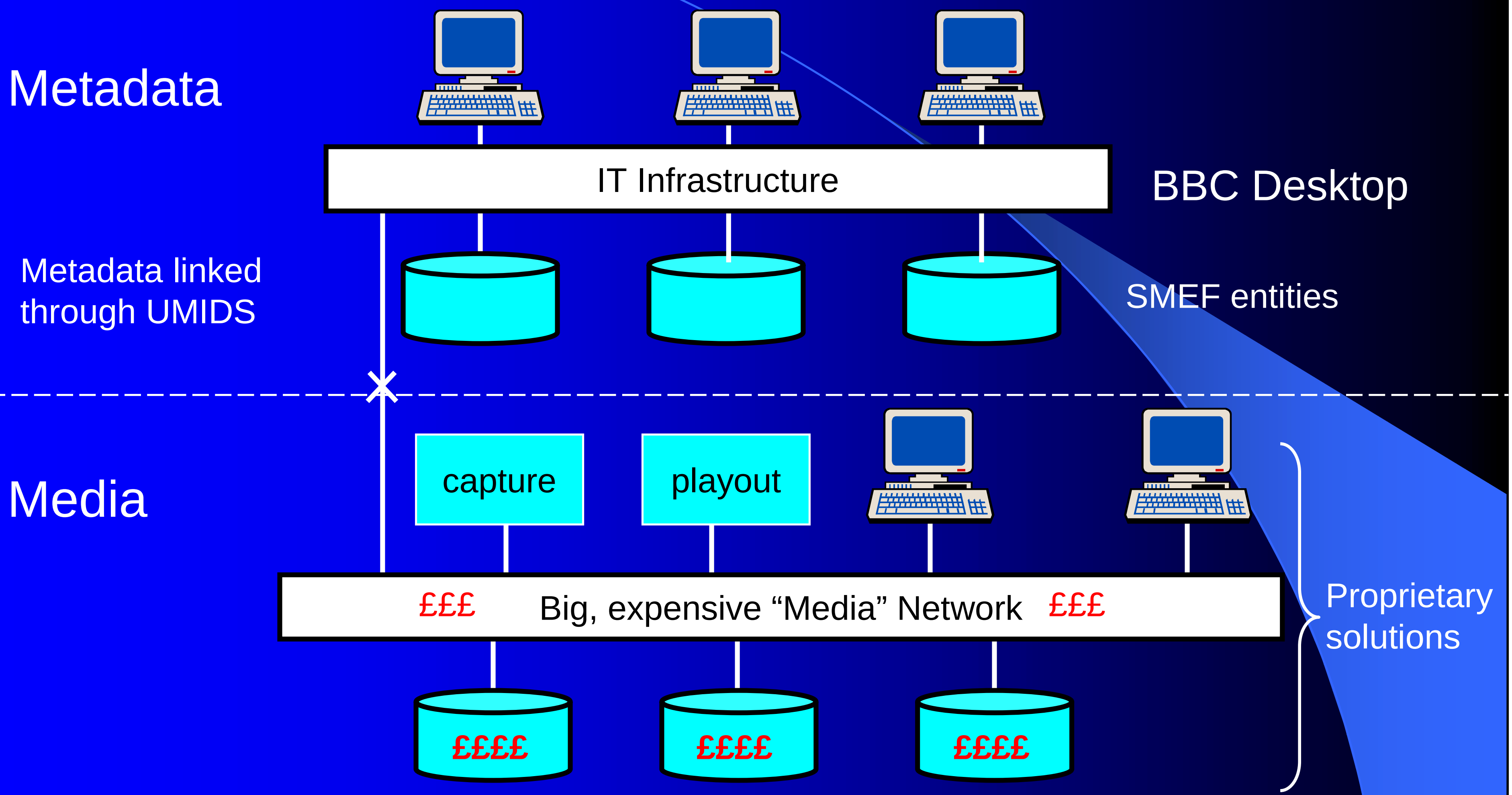
Efficiency improvement requires quick,
and in many cases instant, access to media.
(Tapes are out networks are in?)

INESC PORTO

orbit

BBC R&D

"Media" networks are VERY expensive



INESC PORTO

orbit

BBC R&D

How about the Business Case?

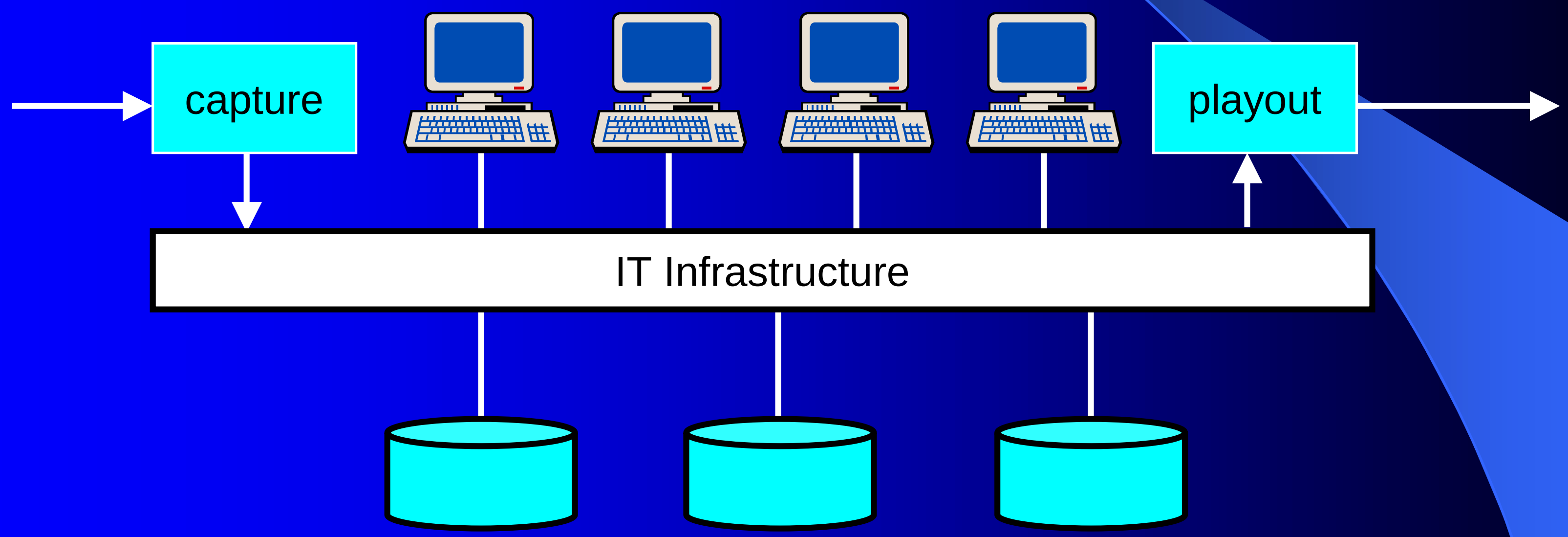
Savings must be greater than costs

The Media Network is the major capital cost

Could this be another job for IT?

ORBIT

Puts media and metadata in one IT network



INESC PORTO

orbit

BBC R&D

Media Handling in an IT network needs:

Efficient media compression

- Production quality for work in progress
- A range of “browse” qualities for different working methods

Efficient storage and transport

- Network and resource management
- Load sharing

Current market offerings will not run on cheap IT kit

But ORBIT does

INESC PORTO

orbit

BBC R&D

Orbit chose:

MPEG2 - Long GoP for "production" quality

- Gives 3 to 1 improvement in storage efficiency

MPEG1 - I frame only, for "browse" quality

- Allows trick modes

These are pragmatic first choices. The system itself doesn't care.

INESC PORTO

orbit

BBC R&D

Long GoP MPEG2 for production quality

- Gives the best quality for a given bit-rate
- And - it's an open standard

But:

How do you:

- Edit to frame accuracy?
- Cascade without quality loss?
- Get from compressed to component and back again?
- Change bit-rate without quality loss?

INESC PORTO

orbit

BBC R&D

ACTS

AC078

ATLANTIC

Advanced **T**elevision at **L**ow bit rates **A**nd **N**etwork **T**ransmission over **I**ntegrated **C**ommunication systems

Participants in the Project:

British Broadcasting Corporation (BBC)	UK
Centro Studi e Laboratori telecomunicazione (CSELT)	Italy
Ecole Nationale Superieure des Telecommunications (ENST)	France
Ecole Polytechnique Fédérale de Lausanne (EPFL)	Switzerland
Electrocraft	UK
Fraunhofer-Institut für Integrierte Schaltungen (FhG-IIS)	Germany
Instituto de Engenharia de Sistemas e Computadores (INESC)	Portugal
Snell & Wilcox Ltd (S&W)	UK

INESC PORTO

orbit

BBC R&D

Achievements of ATLANTIC

- Cascading with zero loss and inter-working between compressed and uncompressed
- SMPTE Standards - the Recoding Data Set (MOLE™)
- Frame accurate editing including mixer effects
- Bit-rate changing with no quality loss
- Transmission and storage on low-cost IT networks and servers

All based on Long GoP MPEG2 and demonstrated at IBC97 and IBC98

Read more at http://www.bbc.co.uk/atlantic/

INESC PORTO

orbit

BBC R&D

Long GoP MPEG2 :

- Reduces storage and transport costs by 60%
- Brings forward the date when IT equipment can do broadcast by 5 at least years
- For most applications, that means now

INESC PORTO

orbit

BBC R&D

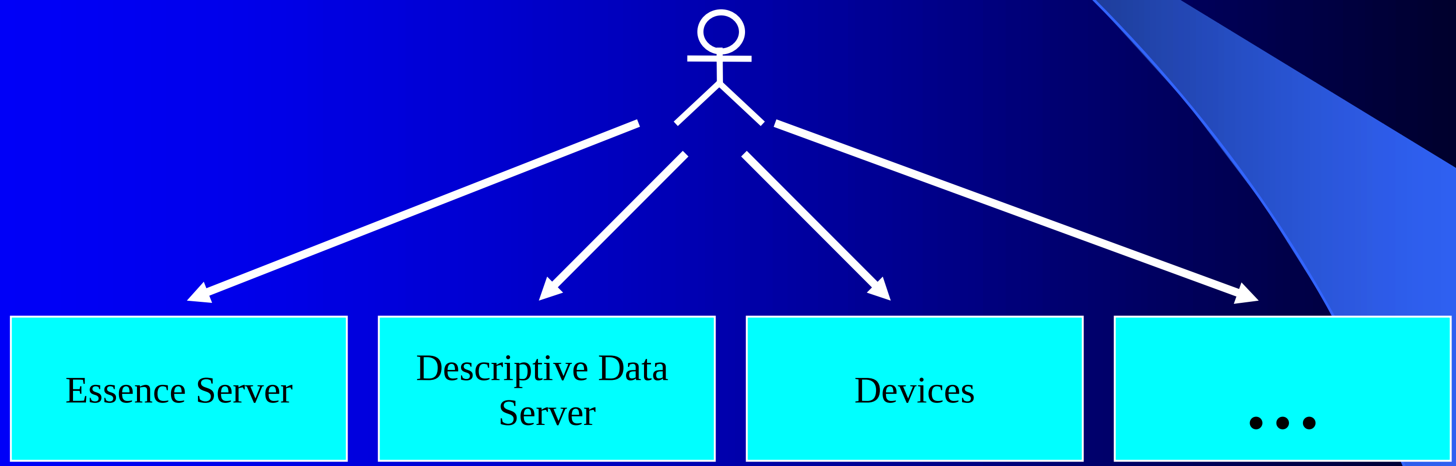
That’s bit-rate out of the way, what on earth is middleware?

INESC PORTO

orbit

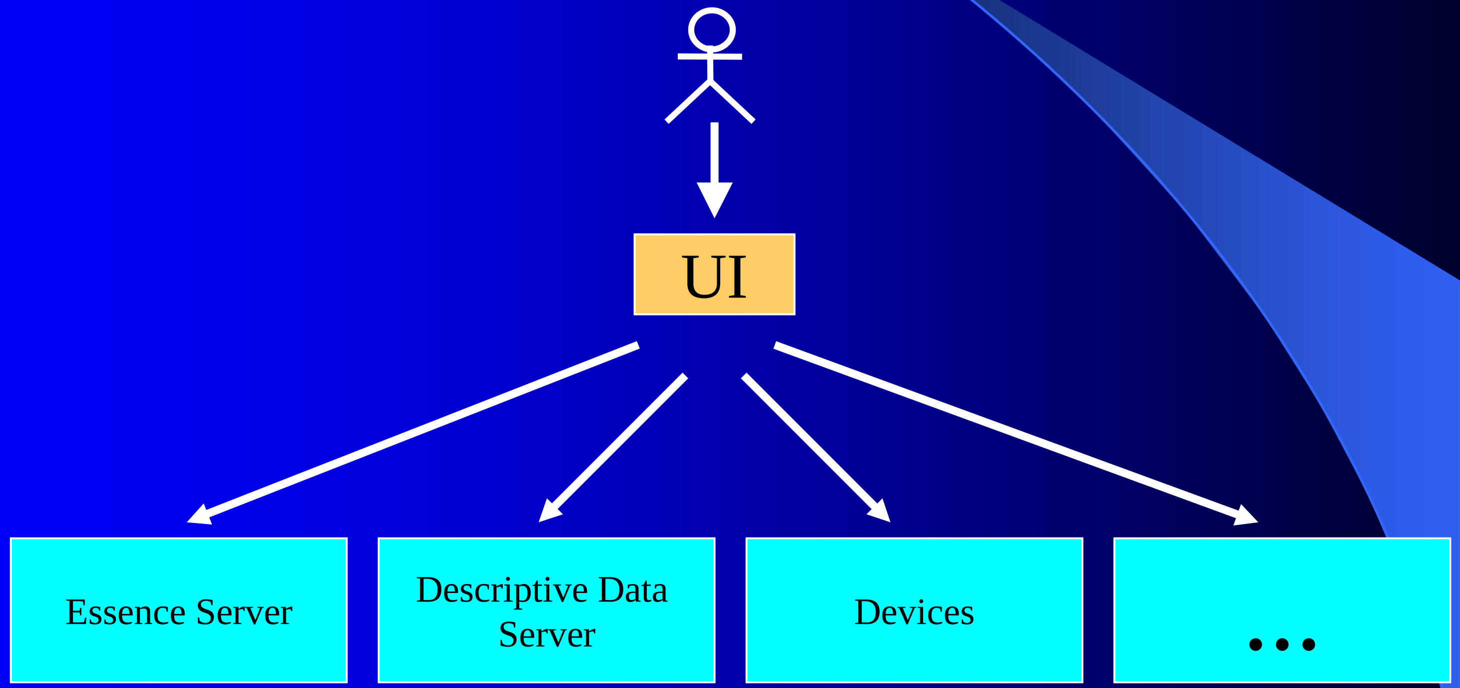
BBC R&D

Manual Integration



R&D

Client Integration

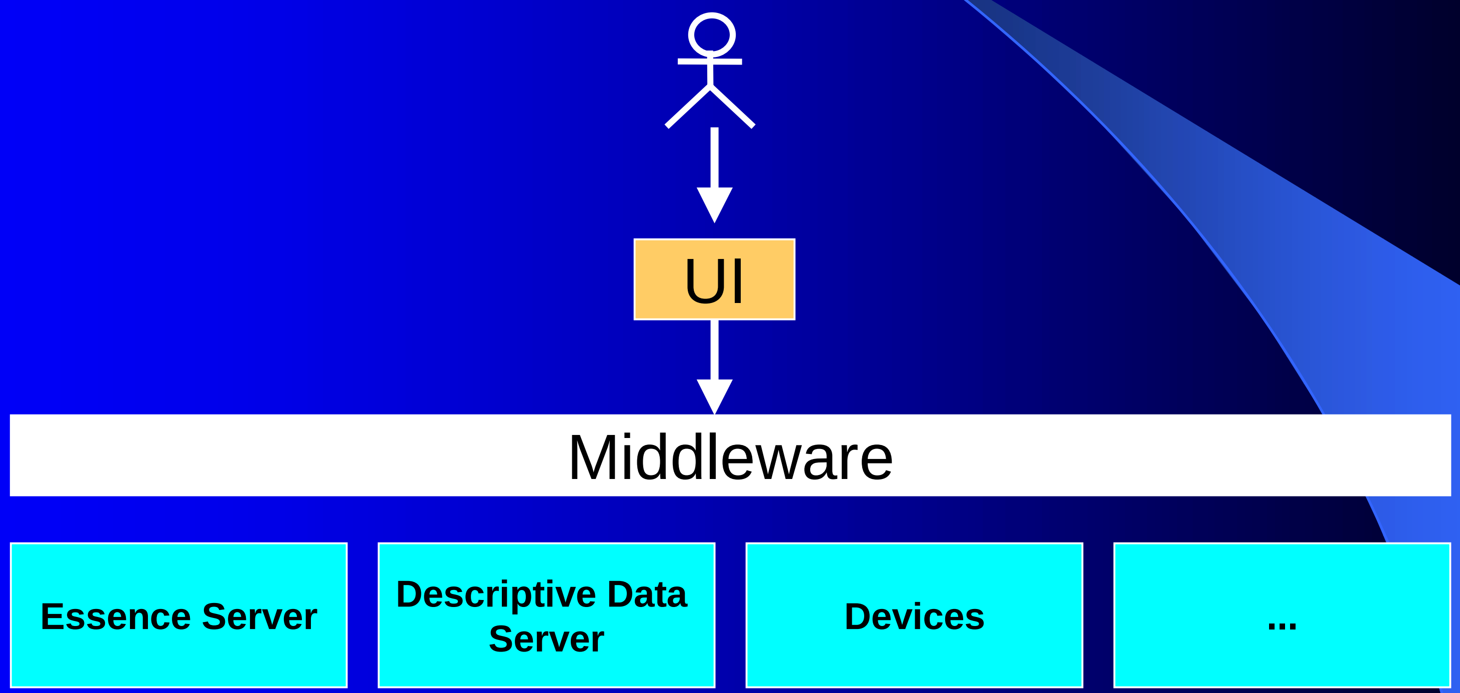


INESC PORTO

orbit

BBC R&D

Middleware Integration



INESC PORTO

orbit

BBC R&D

Existent middleware solutions provide:

Connections across any network

A reasonable ability to inter-work between platforms and systems

Distributed processing for efficient use of resources

Some common interface formats for data

But:

They're designed for relatively small volumes of data

Video is massive, so existing middleware won't work

INESC PORTO

orbit

BBC R&D

ORBIT Uses:

CORBA — For its *implementation*

<?XML!> — For its *information*

CORBA*

Provides a middleware foundation that is:

Platform independent, Intel, Sun, IBM…..

Operating system independent, Windows, Unix, Sun…..

Vendor independent

Standardised and published by the Object Management Group

Implementations available Open Source

*Common Object Resource Broker Architecture

INESC PORTO

orbit

BBC R&D

<?XML!> *

Provides a (meta)data communication format that:

Is standardised by the W3C Consortium

Used and understood throughout the IT industry

Allows the data and its schema to be carried together

XML eXtensible Markup Language

INESC PORTO

orbit

BBC R&D

For its media transfer ORBIT uses:

Advanced Authoring Format

For moving its “work in progress”

Material Exchange Format

For moving its source and finished programmes

INESC PORTO

orbit

BBC R&D

ORBIT has developed its own Broadcast Middleware

DIMICC — For its *media handling*

MBE — For its *business rules*

DIMICC * Provides the media handling

It manages the hardware devices and software services in ORBIT

Transfers the production and “browse” quality essence

Sets up and breaks down network connections as required

Manages the location of media in the network

Manages instantiation and distribution of services

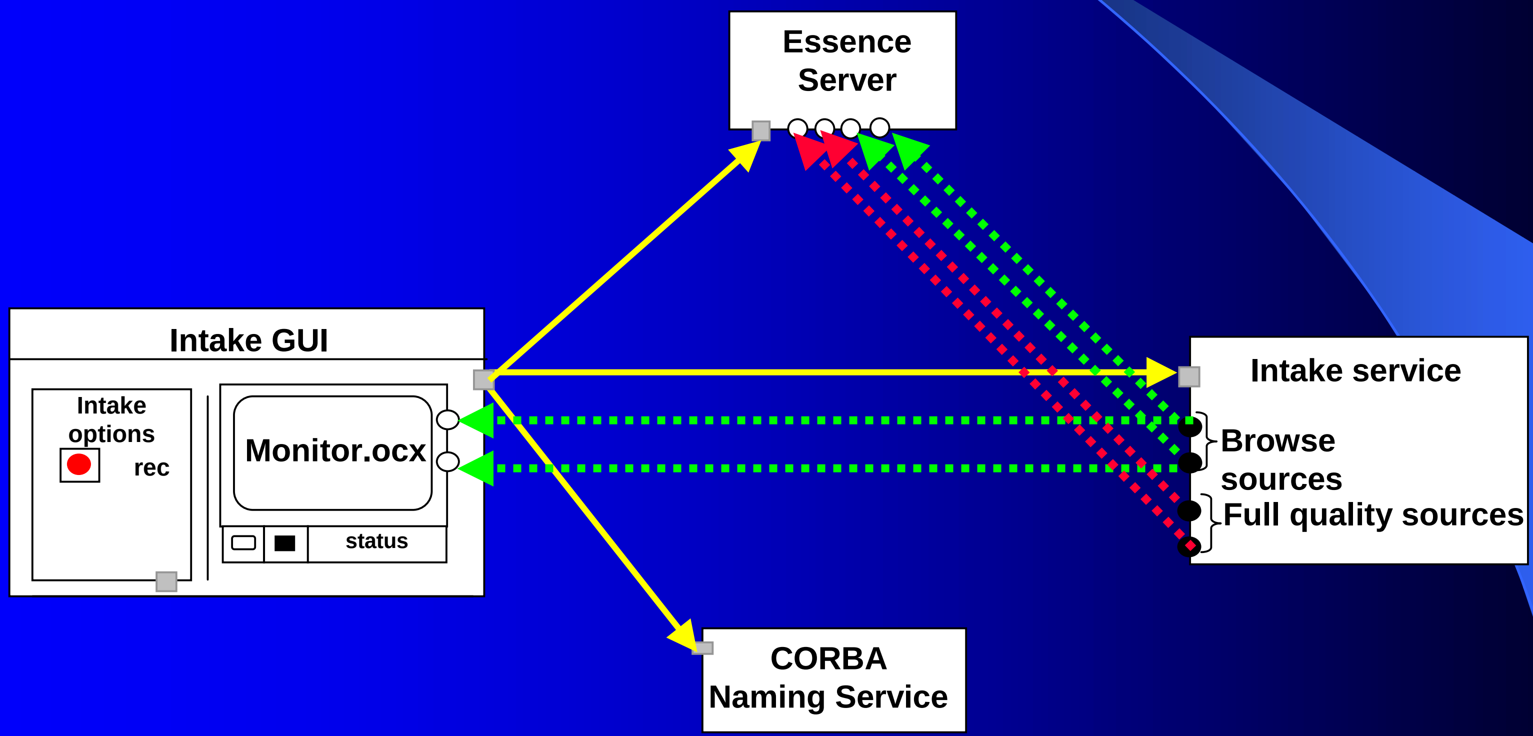
*DIMICC, Distributed Middleware for multimedia Control and Command

INESC PORTO

orbit

BBC R&D

DIMICC – Use Case



INESC PORTO

orbit

BBC R&D

MBE *

Provides the business rules

It brings together the media and devices, using DIMICC, and the descriptive metadata, using XML

XML provides data model independence and allows evolution

It implements the business rules - the who, the where and the when

*MBE - Middleware for the Broadcast Environment

INESC PORTO

orbit

BBC R&D

ORBIT: the bottom line.

Uses commodity IT for metadata *and* essence

Uses best of breed technologies where they exist

Has developed *efficient* Broadcast Middleware

Provides an open API for application developers

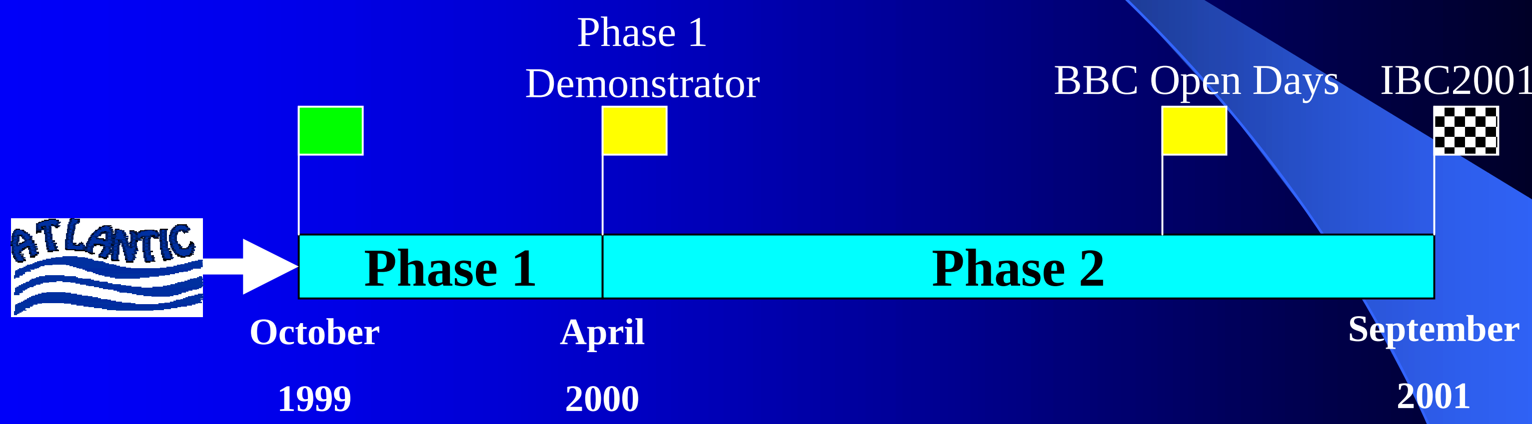
ORBIT is a scalable and affordable broadcast infrastructure

INESC PORTO

orbit

BBC R&D

ORBIT Project Timeline

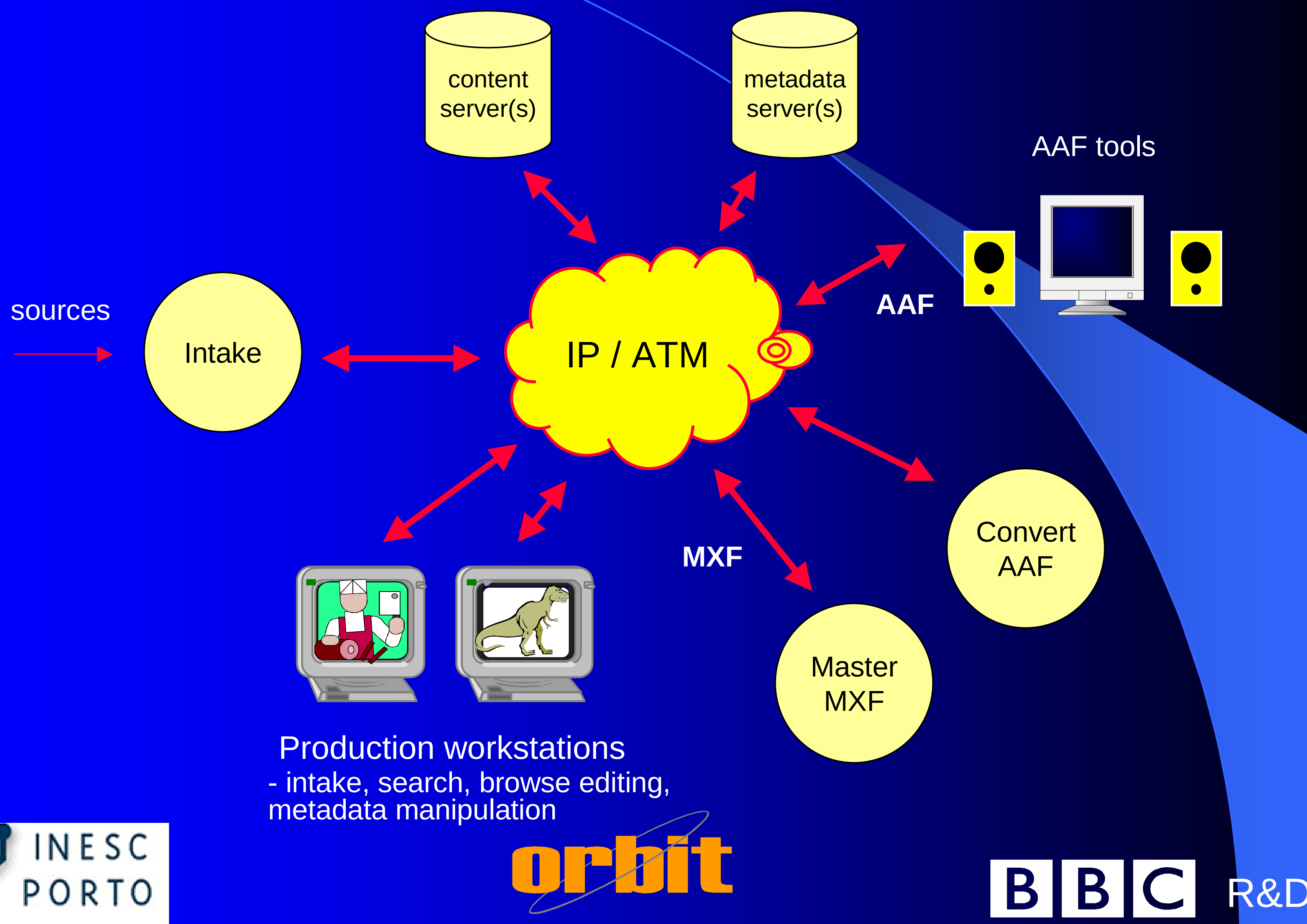


orbit

INESC PORTO

BBC R&D

ORBIT at the BBC Open Days (May 2001)



EXCHANGE FORMAT HANDLING THE MATERIAL (MXF)

Vitor Manuel Teixeira

INESC Porto, Portugal,
vitor.teixeira@inescporto.pt, tel:+ 351222094200, fax:+351222084172

With the introduction of servers and computing devices into the broadcast chain, the exchange of content as a file becomes a pressing need. Much work has been done on the system requirements of the new networked program chain and many goals have been defined, notably by the EBU -SMPTE task force for the harmonisation of standards. An area that needs urgent addressing is the file format, which will be used for the exchange of content between servers. There are a variety of file formats in existence, but none that could claim the title of the "standard" interchange file format. The G-FORS project partners are working hard in association with standards bodies, such as the SMPTE, trade organisations and corporations, such as the Pro-MPEG forum and the Advanced Authoring Format (AAF) Association, to develop and agree upon a format that is simple, flexible and extensible. This paper presents a format that achieves these goals and presents tools to allow creating applications from different vendors that interoperate.

THE G-FORS FILE FORMAT

The exchange of video as files is seen as the key technology that will enable interoperability between different systems. The identification of a simple file format that could handle video, sound and metadata was seen as a vital component for standardisation. G-FORS looked at the various proposals for such a format and decided the likeliest candidate was the Media Exchange Format (MXF) that is being formulated by the Pro-MPEG Forum and the AAF Association. G-FORS wanted a file format that would meet the needs of simple interchange as well as being resolution and compression independent. The partners in the G-FORS project actively adopted the MXF file format specification for their project. They are implementing and using the specification to ensure wide adoption of the Operational Patterns defined in the final MXF standard. MXF provides a 'wrapper' for signal interfaces and disk-based storage of television images, sound, data and associated metadata. Its foundations rely on the SMPTE KL V coding specification (SMPTE 336M -Data Encoding Protocol using Key-Length- Value) that allows full flexibility and extensibility. A compliant MXF stream defines a base dictionary and its organization in a set of templates so that applications are able to wrap and unwrap essence and metadata in a common space.

THE G-FORS SDK

The SDK (Software Development Kit) is divided into two layers allowing developers to use either low-level functions, to access a stream based on the KL V structure with several levels of nested hierarchies, or high-level functions, to navigate in the metadata through a DOM (Document Object Model) structure and retrieve the encoded audiovisual data independently of its coded format. Currently the SDK is not required to decode the essence by means of plug-in codec architectures, still further work will allow its integration with other frameworks that provide this feature. This SDK contains test applications that allow developers to parse, dump and debug files and wrap and unwrap metadata and essence.

CONCLUSION

The G-FORS project forms part of the European Commission Information Society Technology Research program (1ST). Information about the project can be obtained at www.g-fors.com. The ideas coming out of the G-FORS Project will be communicated to standards bodies such as the EBU and SMPTE. A core element of the project will be to build a demonstration system to show the benefits of file transfer using a generic format. The SDK generated within the project will allow third parties to create applications abstracted from the underlying stream syntax and organized in a flexible way.

HANDLING THE MATERIAL EXCHANGE FORMAT (MXF)

Vitor Teixeira

INESC Porto

DASE'2001 Symposium, June 19 – 20, 2001

Overview

- G-FORS project
- File Format
  - What is MXF ?
  - What does it do ?
  - How does it do it ?
  - Basics about MXF
  - MXF structure
- SDK – Software development kit
  - How it is organized
  - Functionalities
  - How to integrate with real applications

INESC PORTO

G-FORS

G-FORS

What is G-FORS?

- Generic Format for Storage
- EC-funded project - Information Society Technologies programme

"Reduce the cost of European programme production by making content available in an efficient and cost effective way"

- 7 partners - commercial, public and research

BBC R&D; CRIL Technology; Enertec; INESC Porto; Philips DVS; Snell and Wilcox; THOMSON broadcast systems

www.g-fors.com

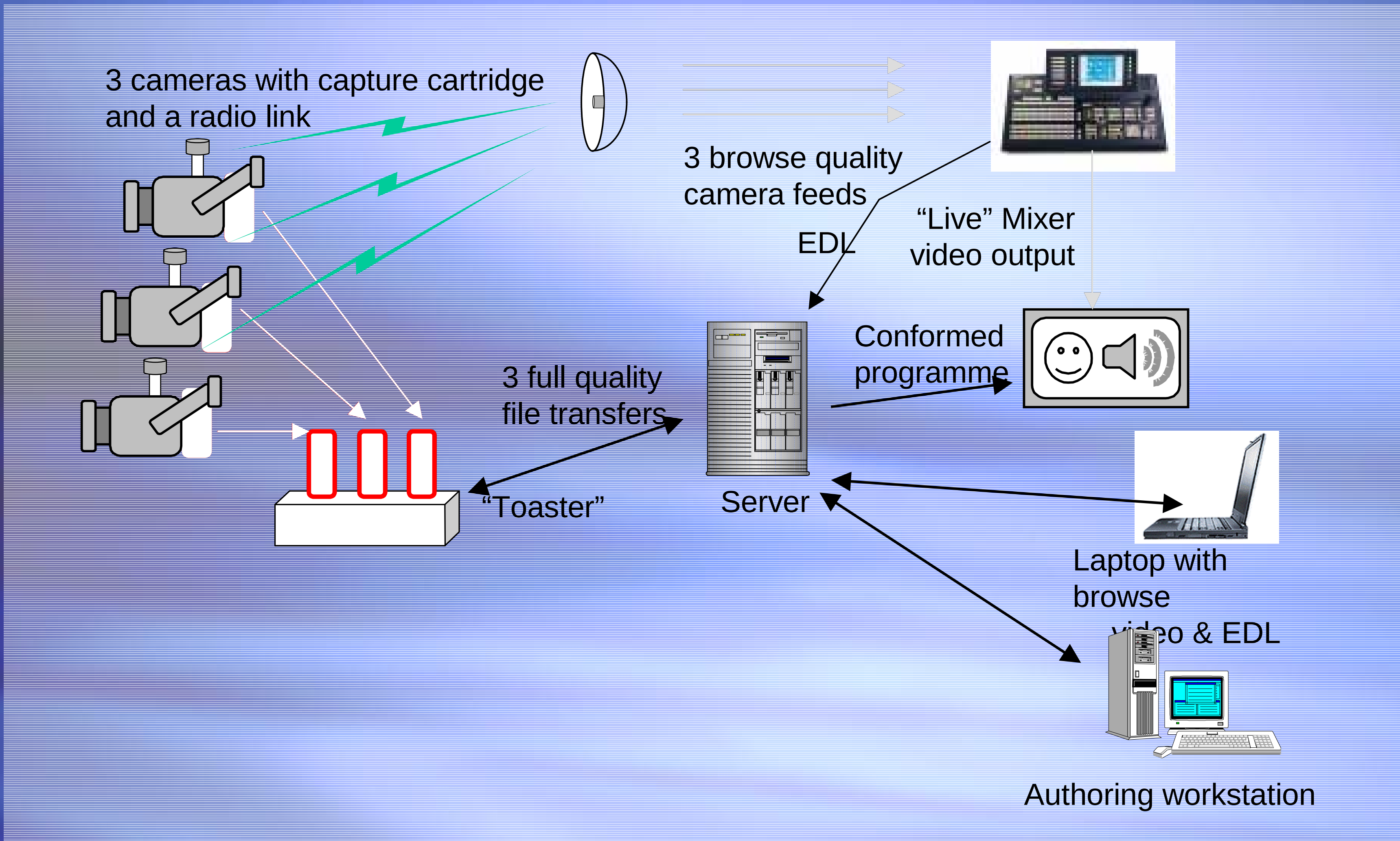
INESC PORTO

G-FORS

G-FORS

- Implemented the file format that accommodates the broadcaster needs
  - simple interchanges (tape replacement)
  - simple editable package (cut edits only)
- Uses a cost-effective infrastructure IT based
- Delivers several flavors of essence integrated with metadata:
  - Accommodate DV and MPEG within the same file format
  - Guaranty a common metadata set of structure and descriptors
- Faster than real time transfers
- Demonstrates possible integration with commercial partners outside the consortium and interoperability
- Is has adopted MXF and is actively working with Pro-MPEG to make it a successful file format.

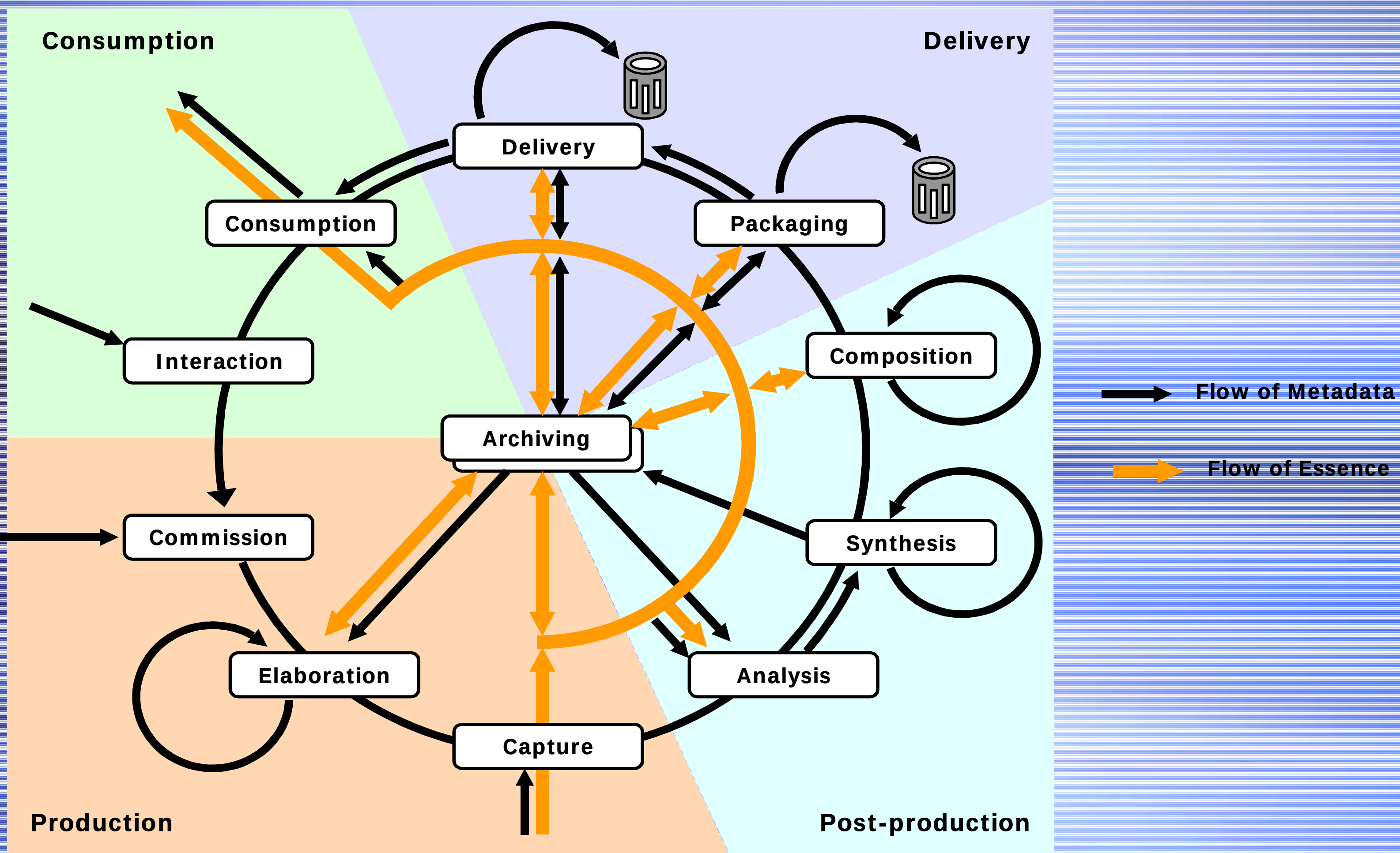
G-FORS demo scenario



INESC PORTO

G-FORS

An Introduction – producing Essence and Metadata



INESC PORTO

G-FORS

What is MXF ?

- An interchange file format to be used within the broadcast chain
- An extensible file format
- A compression agnostic file format
- A versatile file format
- A metadata aware file format
  - Structural metadata
  - Descriptive metadata
- A stream-able file format
- NOT an authoring format
  - MXF allows editable packages with simple cuts
  - Complex audiovisual transformations and effects done with AAF

What is MXF ?

- Over 2 year of hard work by a joint team:
  - Pro-MPEG Forum in association with the AAF association
    - Over 130 members, among the most active BBC, SONY, Snell & Wilcox, AVID, SGI, INESC Porto,…
- Specification divided into a set of documents
  - Part 1: Engineering guidelines
  - Part 2: Format Specification
  - Part 3: Operational Patterns
  - Part 4: Descriptive Metadata sets
  - Part 5: Body formats
- These have all been submitted to SMPTE for ballot (April 2001) and is open to the public (www.pro-mpeg.org)

What does it do ?

- MXF provides …
  - an extensible framework for interchanging Metadata and Essence
  - independence from compression formats
  - a variety of operational patterns to fit different applications
  - a means of encapsulating structural & descriptive metadata
  - a means of relating the metadata with the essence
  - low level file structure for efficient storage and parsing
  - a means of indexing content for random access
  - a stream-able file format for real time contribution

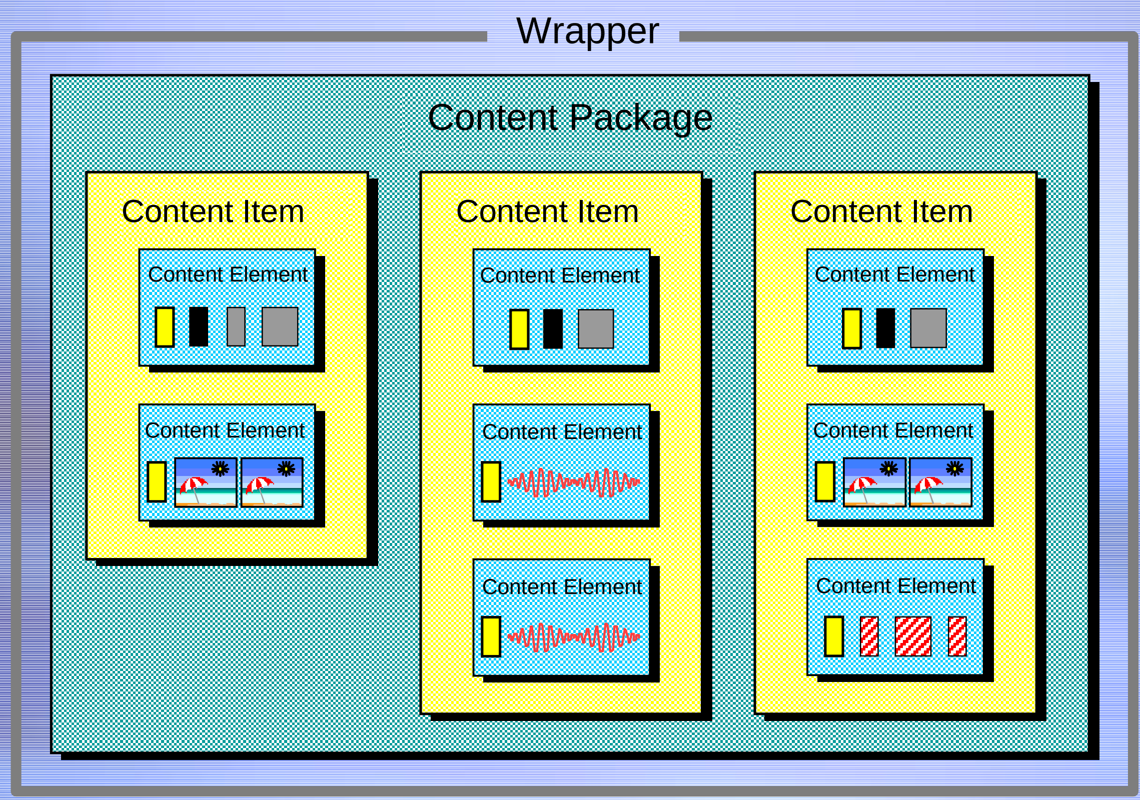
MXF concepts

- Wraps: Essence and Metadata
  - Essence Agnostic
  - Metadata defined by SMPTE, MPEG-7 and the p-META group
- Key-Length-Value Coding (KLV – SMPTE 336M)
- Public dictionary (Registries)
- Uses Unique Media Identifiers (UMIDs) to locate and label essence

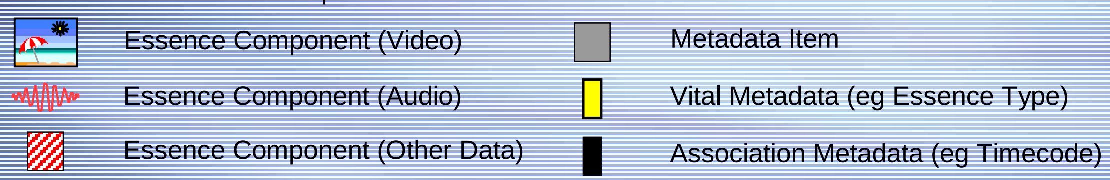
INESC PORTO

G-FORS

Wrapping the essence and Metadata



These are all Content Components:

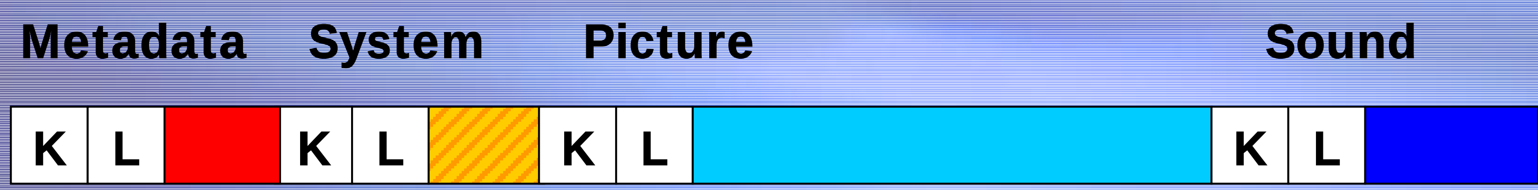


INESC PORTO

G-FORS

Base infrastructure for MXF

- KLV coding
  – Key: a unique identifier
  – Length: how long is the field
  – Value: what is the value of the field



INESC PORTO

G-FORS

The KLV coding scheme

Key	Len	Value

KLV

06 0E 26 34 01 01 01 02 01 05 08 00 00 00 00 00	18	47 6F 6E 65 20 57 …

Not to scale

Dictionary (Local)

Key	Name	Type	Meaning

Lookup

MXF Dictionary (Public)

Key	Name	Type	Meaning
01 05 08 00 00 00 00 00	Version Title	ISO 7 bit	The Version Title
01 04 07 01 00 00 00 00 00	Edge Code Title	ISO 7 bit	Film Edge Code

INESC PORTO

G-FORS

The KLV coding scheme

Key	Len	Value
06 0E 26 34 01 01 01 02 01 05 08 00 00 00 00 00	18	**Gone With the wind**

KLV

Not to scale

Dictionary (Local)

Key	Name	Type	Meaning
01 05 08 00 00 00 00 00	**Version Title**	**ISO 7 bit**	**The Version Title**

Lookup

MXF Dictionary (Public)

Key	Name	Type	Meaning
01 05 08 00 00 00 00 00	**Version Title**	**ISO 7 bit**	**The Version Title**
01 04 07 01 00 00 00 00 00	**Edge Code Title**	**ISO 7 bit**	**Film Edge Code**

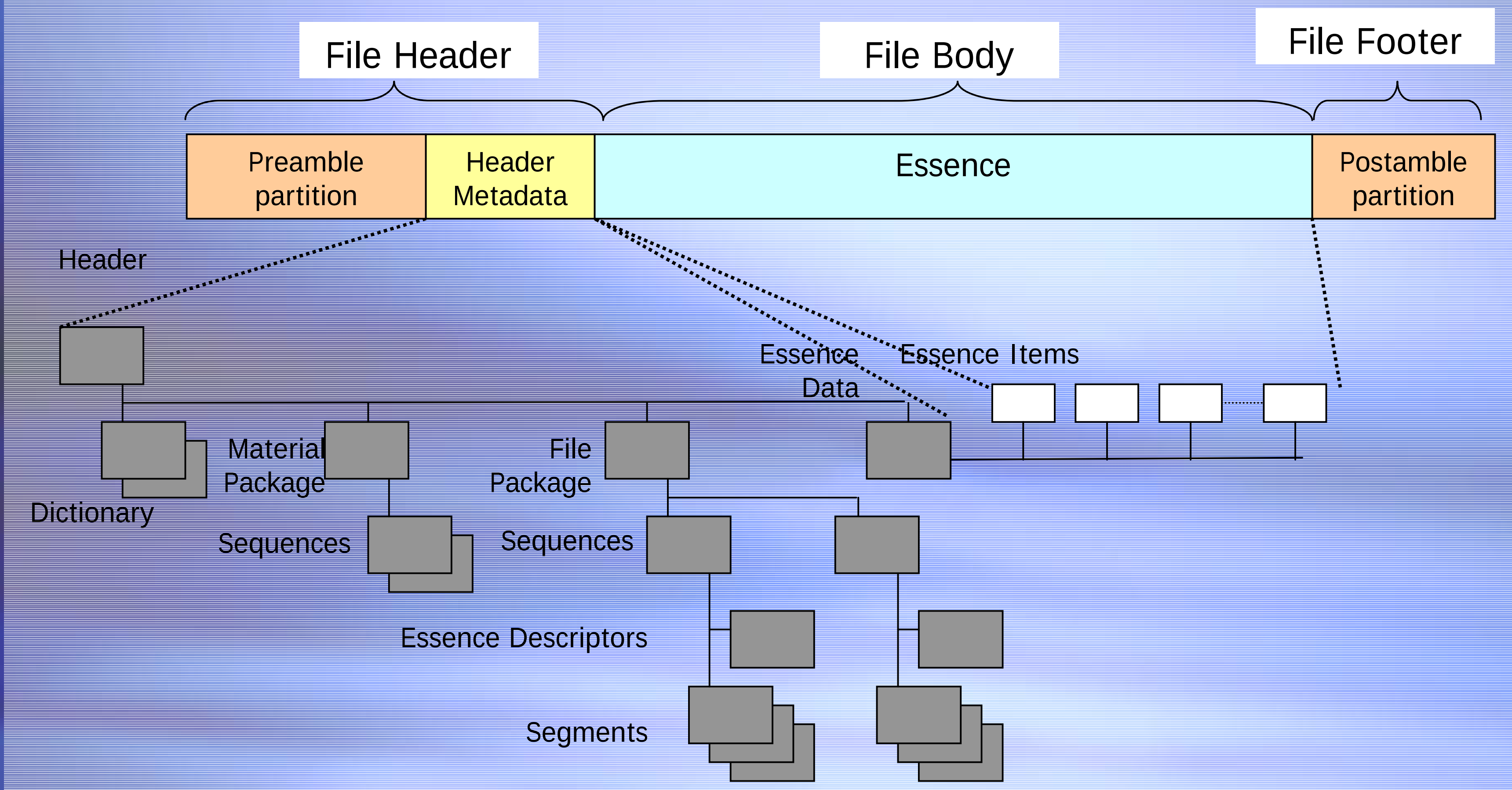
INESC PORTO

G-FORS

Dictionary for the Keys (registry)

- Systems can share common metadata
- Common repository for definitions
- Systems can be extended either in terms of new metadata or new kinds of essence
- Vendors can add their on features

MXF file structure (basic)

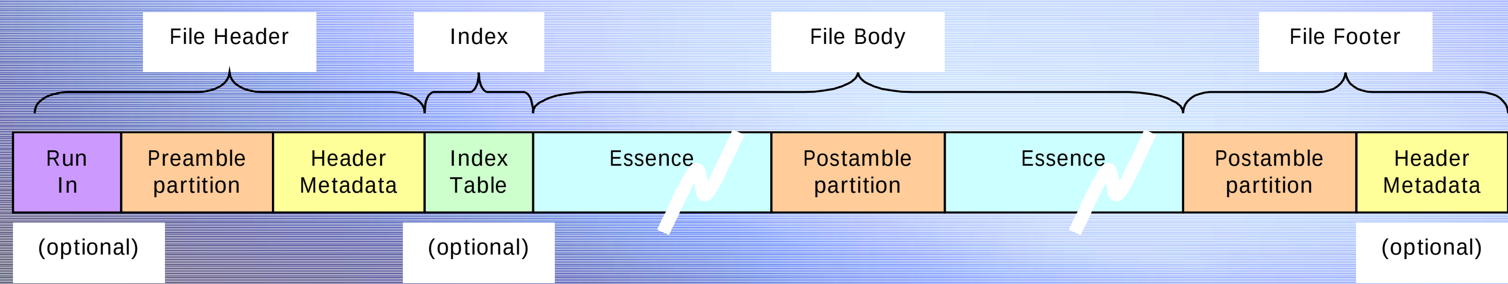


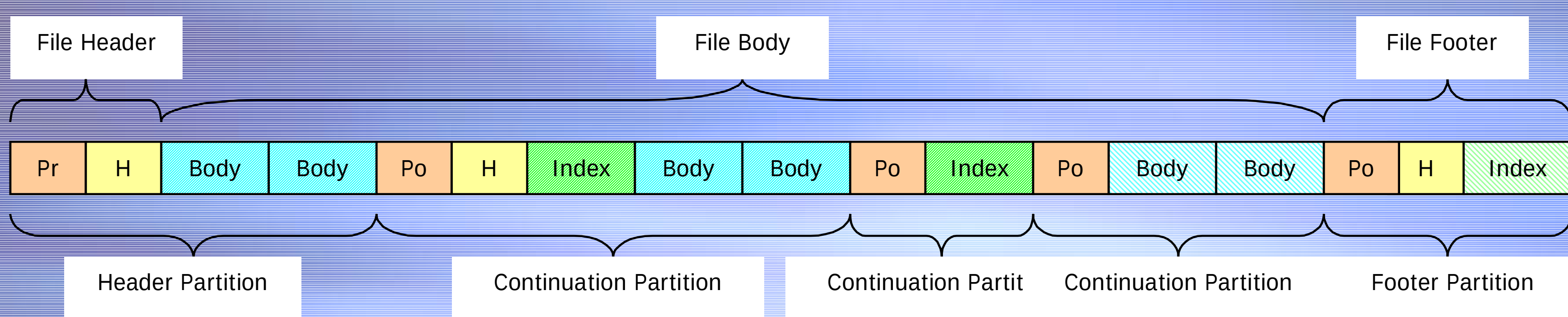
INESC PORTO

G-FORS

MXF file structure (Complex)

- Multiple essence, index tables, run-in, repeated header metadata





INESC PORTO

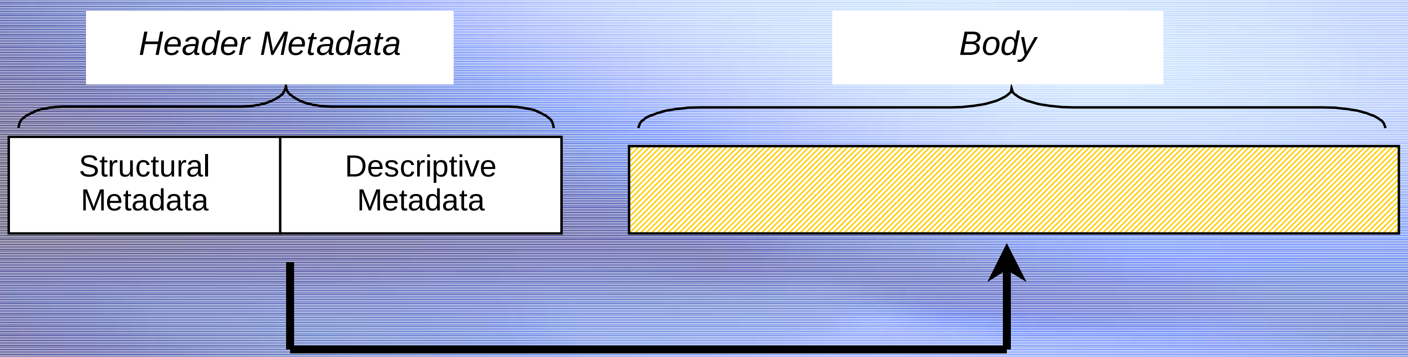
G-FORS

Compression Independence

- Different MXF body types can be KLV wrapped
- Body types can be single essence or multiplexed
- Body type is signaled in the first few bytes
  - Enables early success / failure when streaming
  - Allow rapid identification of body types
- Metadata can be parsed even if essence type cannot be decoded
- Store & Forward devices can report compression type

Operational Patterns

- Operational Pattern 1

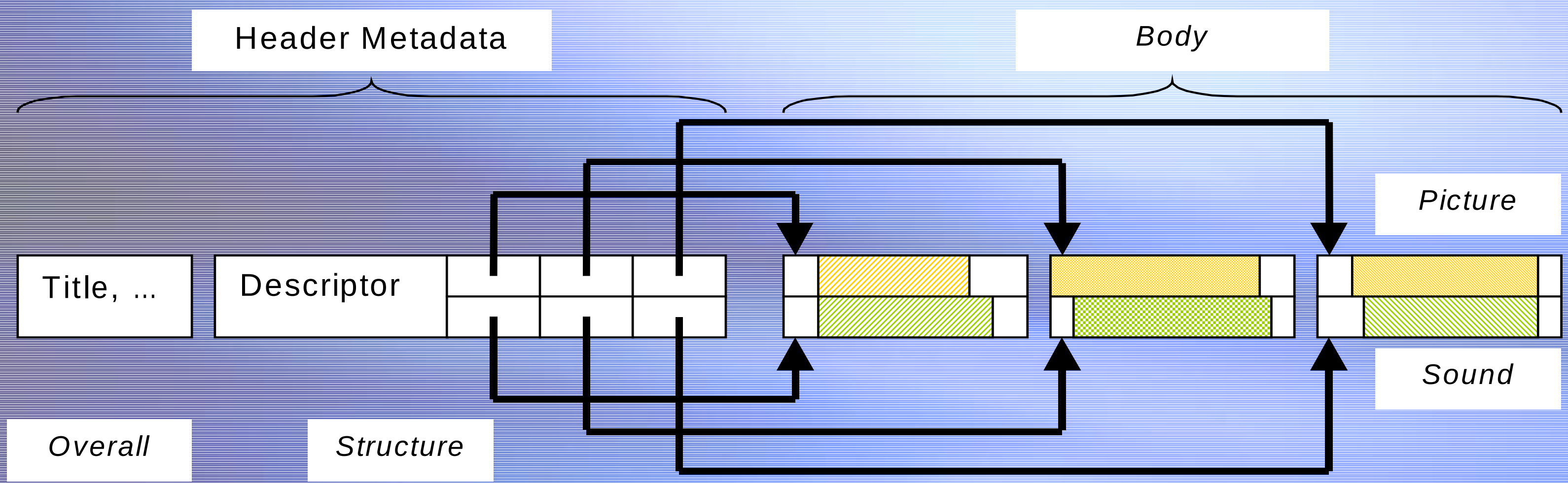


INESC PORTO

G-FORS

Operational Patterns

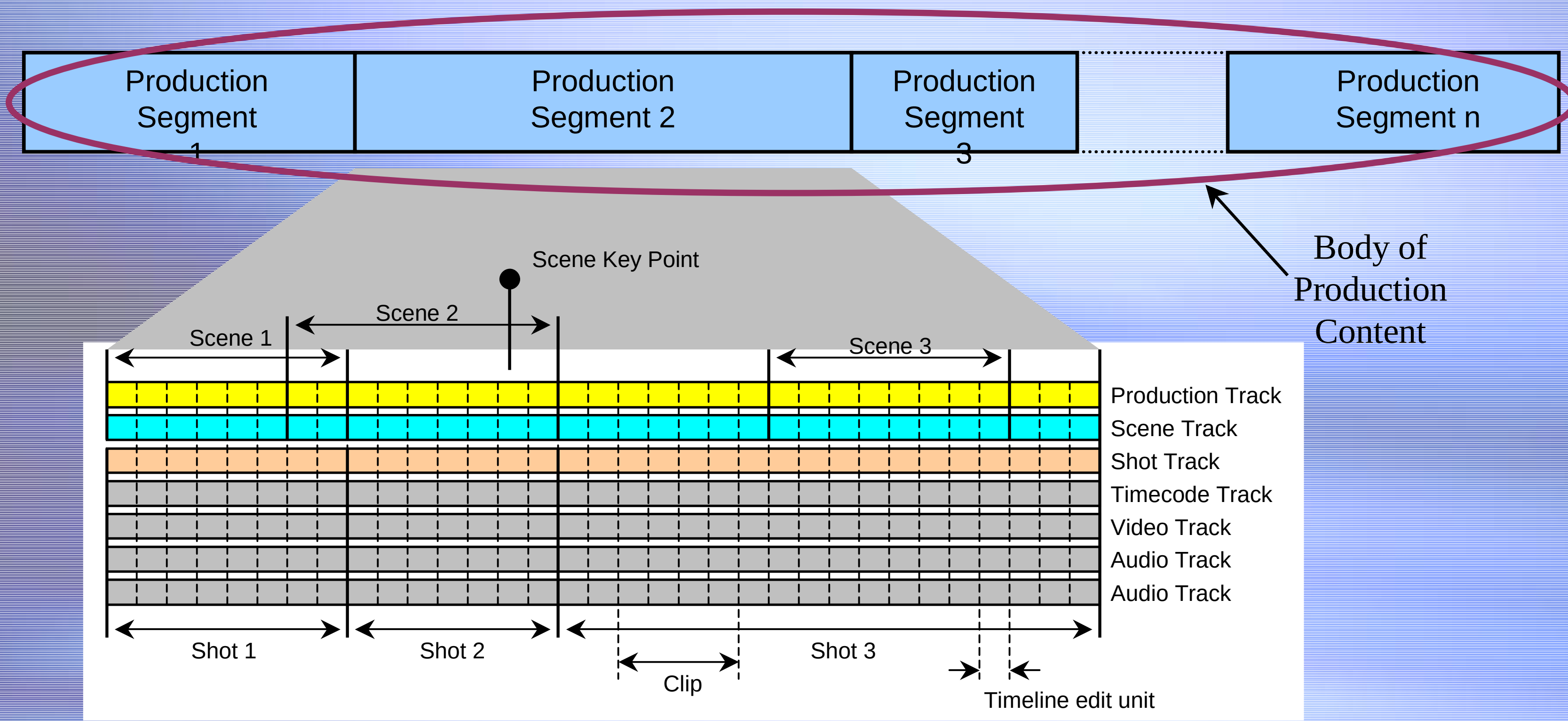
- Operational Pattern 5



INESC PORTO

G-FORS

Header Metadata



INESC PORTO

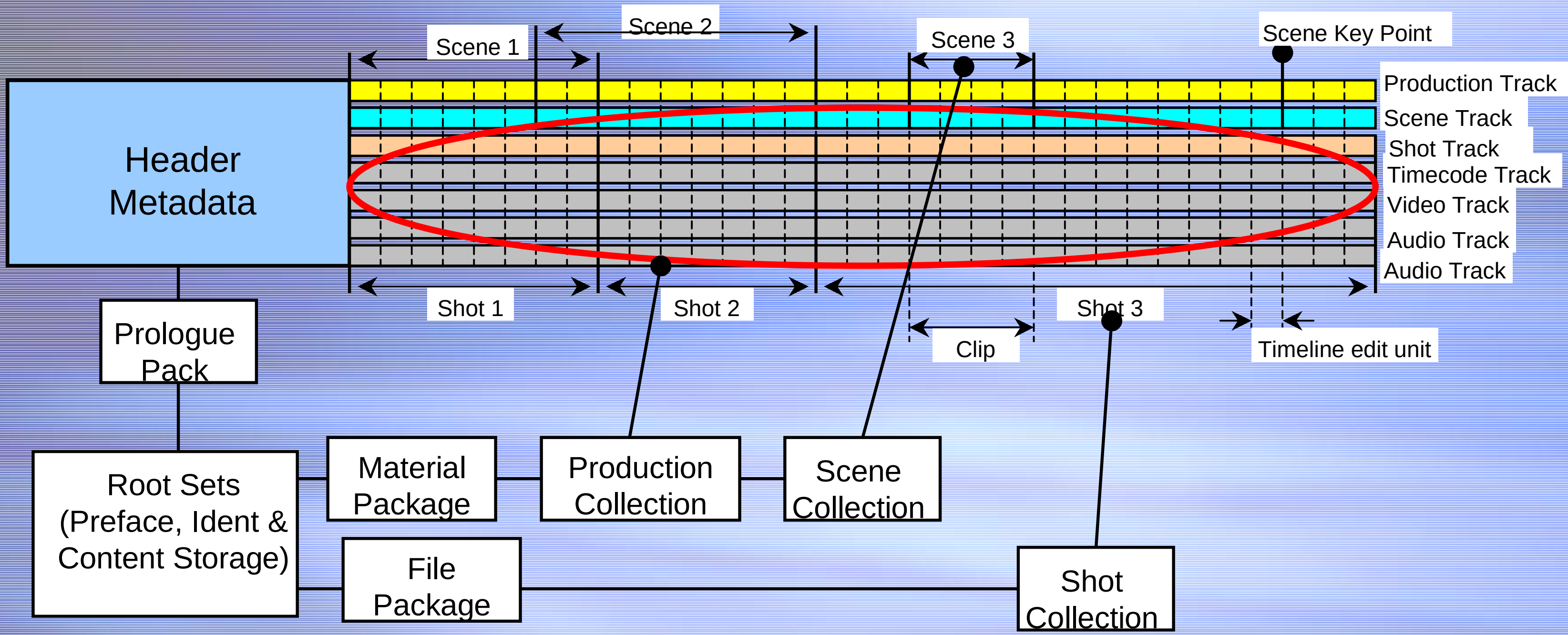
G-FORS

Structural Metadata

- Major elements such as “byte order” in the file
- UMIDs for the essence components
- Packages
  - A group of tracks
  - Material package defines the “output” timeline
  - File packages define the “input” timelines
- Tracks defined so far
  - Timecode, Video, Audio, Events
- Sequences of Segments
  - i.e. how the video “clips” are ordered and fit together

Descriptive Metadata

- Depends on the use of the file
- Current set is aimed at creating Broadcast Programs
  - Production metadata: Titles, episodic information
  - Definition of scenes, shots, participants, awards

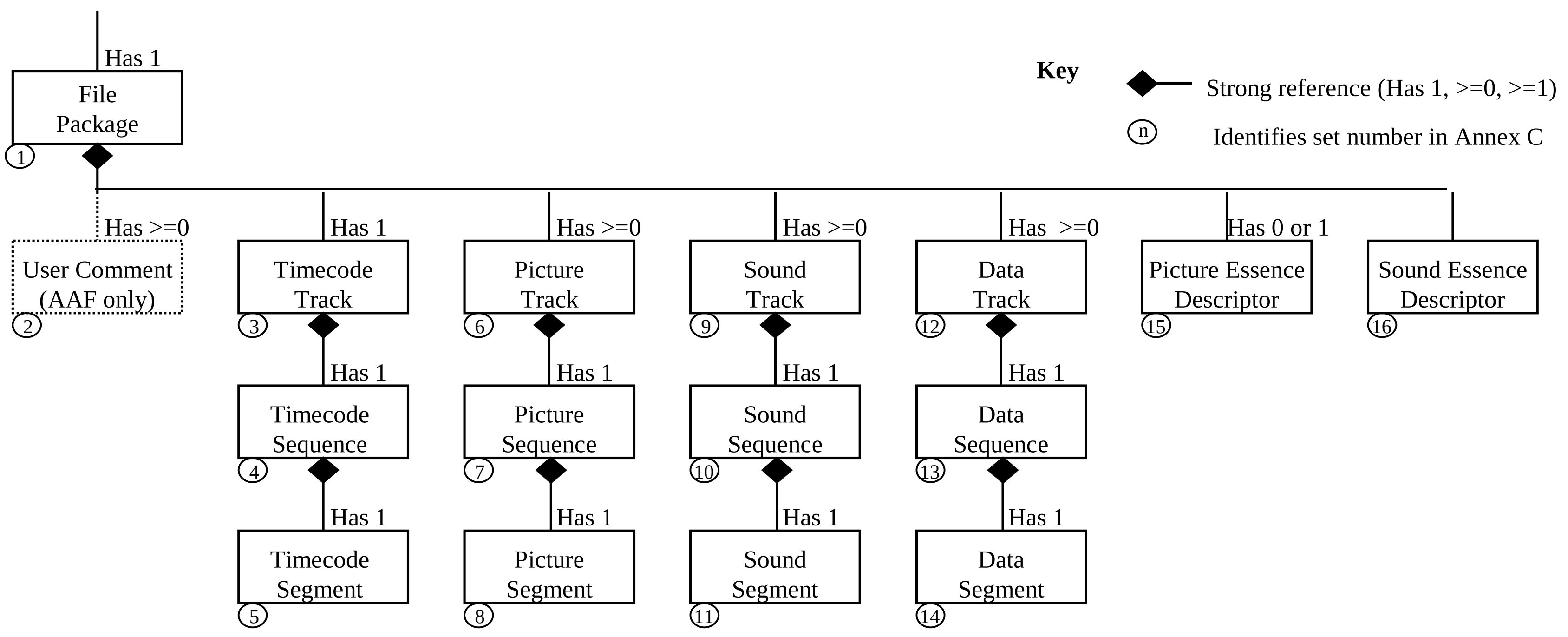


INESC PORTO

G-FORS

How it is structured and defined?

- Defined in Unified Modeling Language (UML)
- Public dictionary will also be released in XML form

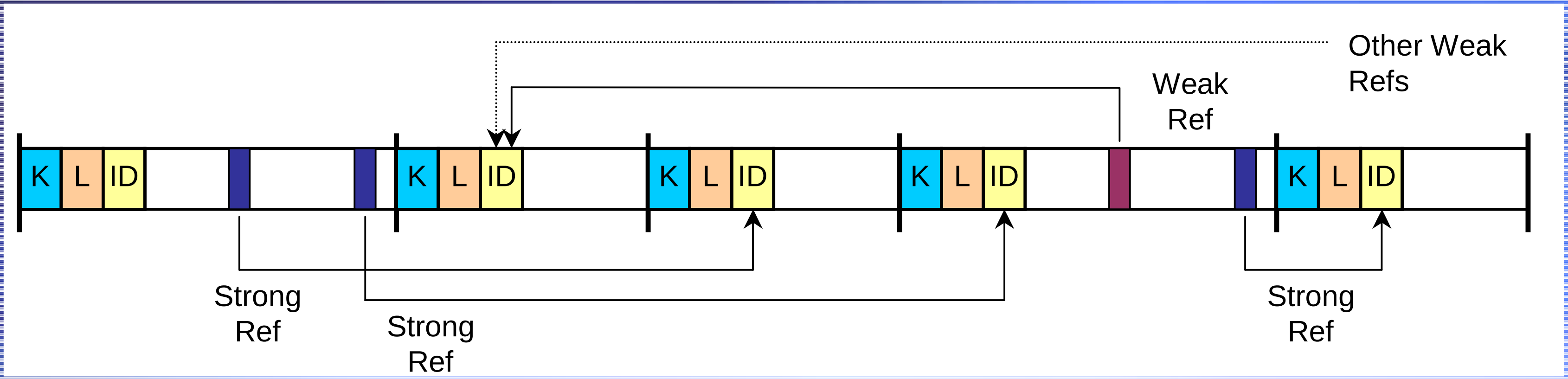


INESC PORTO

G-FORS

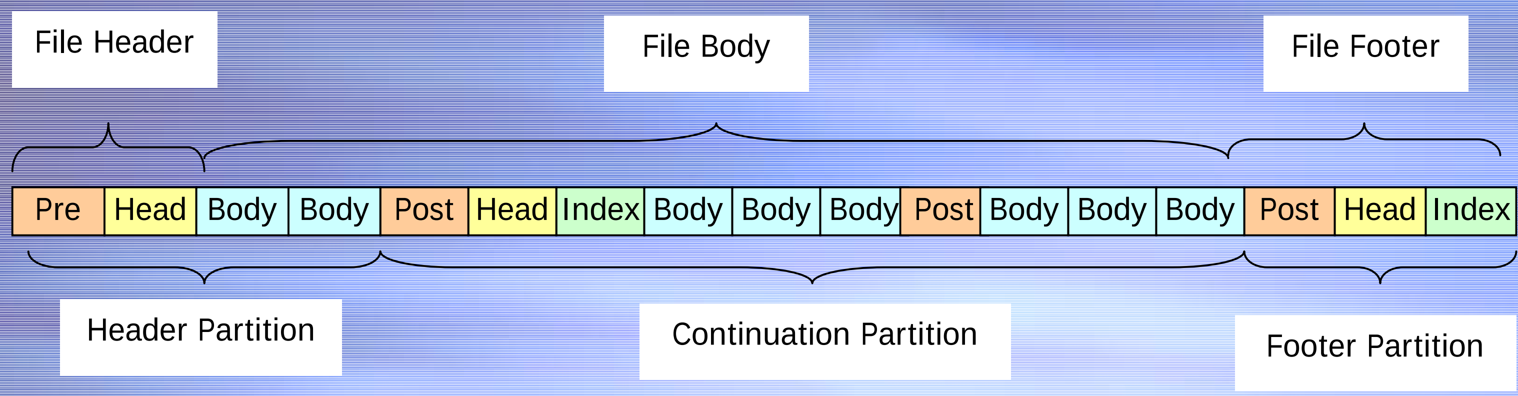
How to connect metadata sets

- Referencing of one set to another
  - UUIDs and UMIDs are used as the links
  - strong reference means one to one relationship
  - weak reference means one to many - both are used



Low level structure: Partitions

- Divides file into partitions containing a single “thing”
- Partitions have an integer number of sectors
- Sectors are a defined size (default sector size: 4096)
- The order of elements in a partition is defined
- Ease the indexing process …

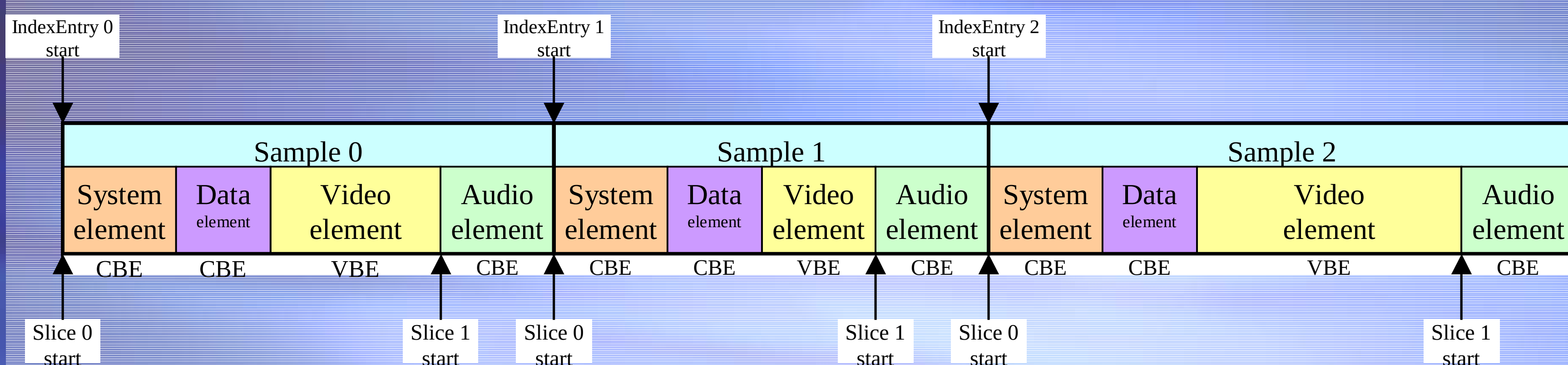


INESC PORTO

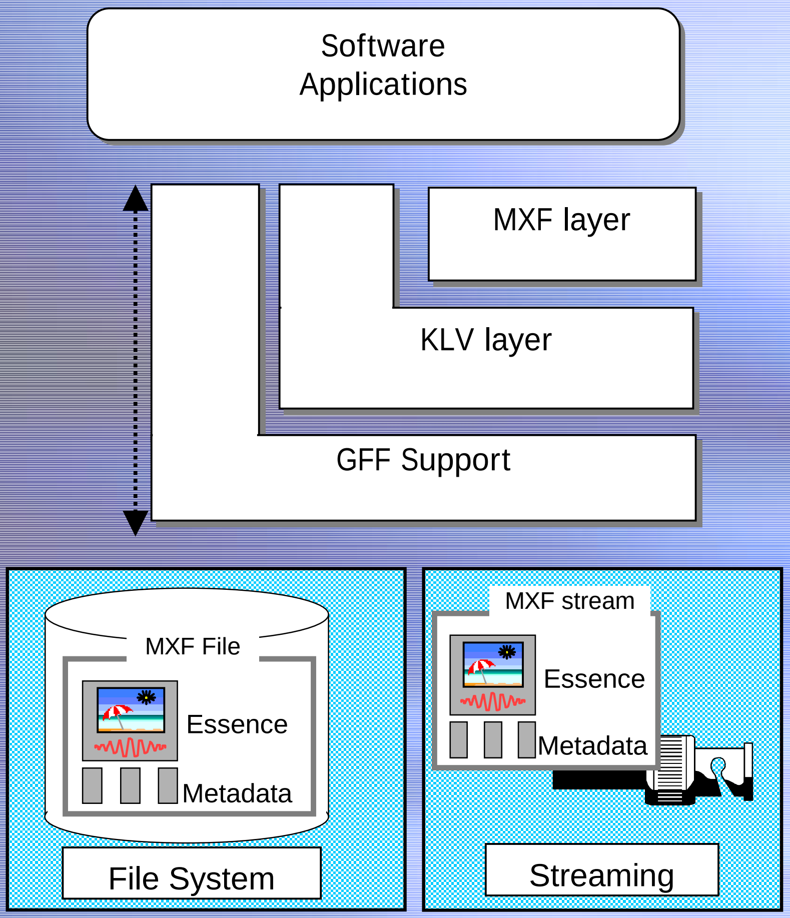
G-FORS

Indexing Content

- Generic Index Table solution
  - copes with CBR and VBR
  - copes with Temporal re-ordering
  - cope with predictive compression schemes
  - simple implementation for simple body types



Handling MXF



Levels of access from the application

L3 – High level – access file as individual MXF persistent objects

L2 – Mid level - KLV layer access file as KLV items either to read or write

L1 – Low level – abstraction from storage device

L0 - Low level – basic access to I/O routines

INESC PORTO

G-FORS

MXF SDK

- SDK is a set of objects stored in a library that implement all 3 layer functionalities
- Implementation done in C++;
- ANSI/POSIX compliant;
- High performance;
- Multi-platform, currently tested in:
  - Win xx: as a dynamic library (DLL)
  - Solaris / Linux: as a static library (although in future implementations can evolve to a dynamic lib)
- SDK developed in 3 phases:
  - Low-level API (Transparent access to files, cartridges, network,…);
  - Mid-level API (KLV-SMPTE336M compliant);
  - High-level API (MXF Compliant);

INESC PORTO

G-FORS

MXF SDK (Mid-level API)

- SMPTE 336M compliant
- KLV:
  - Possible to parse/decode KLV data:
    - Metadata
    - Essence
  - Possible to encode essence into a KLV stream
- In a MXF application point of view
  - Have to translate the key into a meaning (later to be implemented by the MXF SDK high level)

INESC PORTO

G-FORS

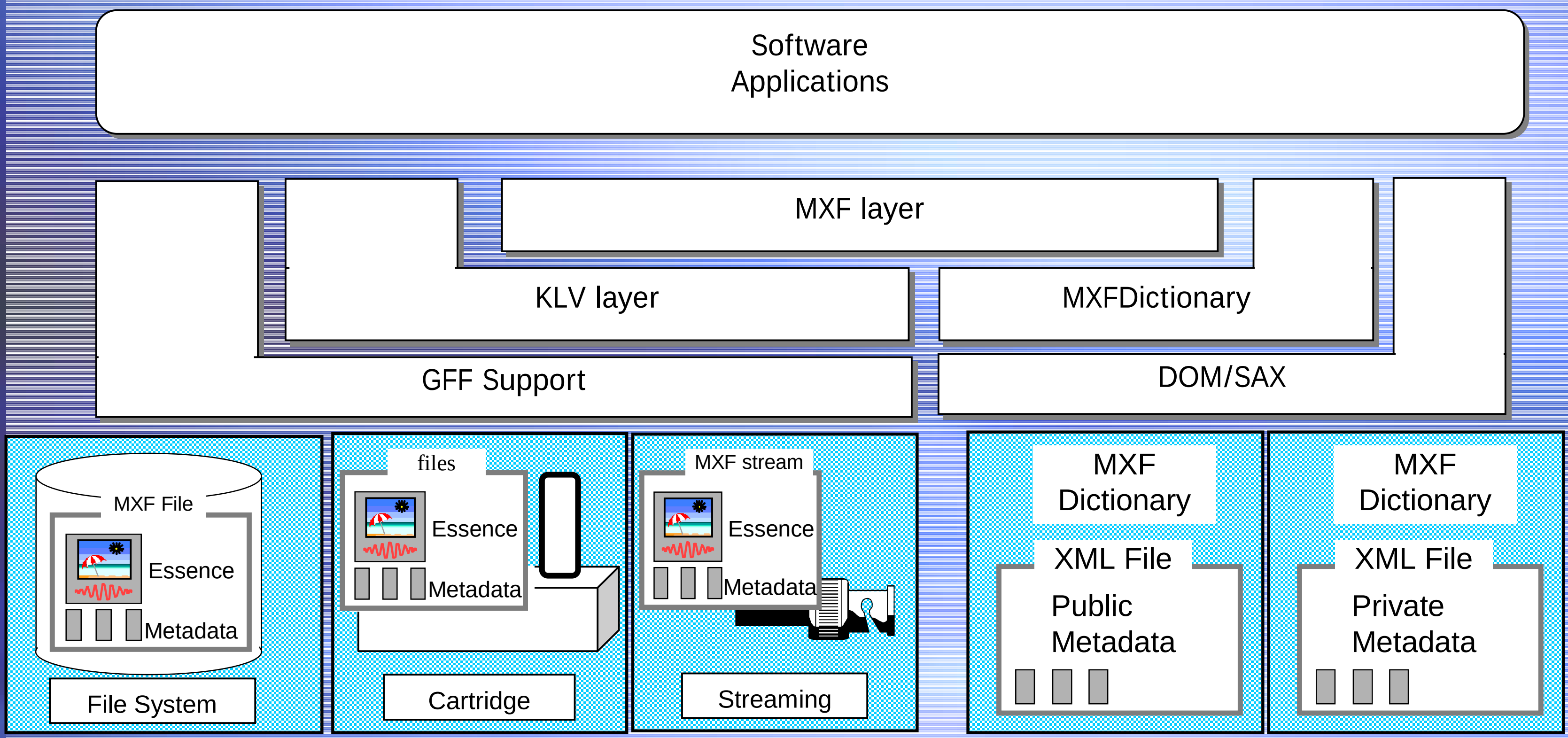
MXF SDK (High-level API)

- Instantiate the metadata objects as well as serialized them to the file;
- Navigate through the node objects (equivalent to the DOM API – Document Object Model)
- Possibility to follow the links (weak and strong references) as if physically KLV nested
- Enhanced to read essence from the partitions
- Possibility to interpret MXF data depending on the loaded dictionary;
- Events schemes to signal update on metadata;
- Additional layer that loads dictionaries written in XML form;

INESC PORTO

G-FORS

MXF SDK stack

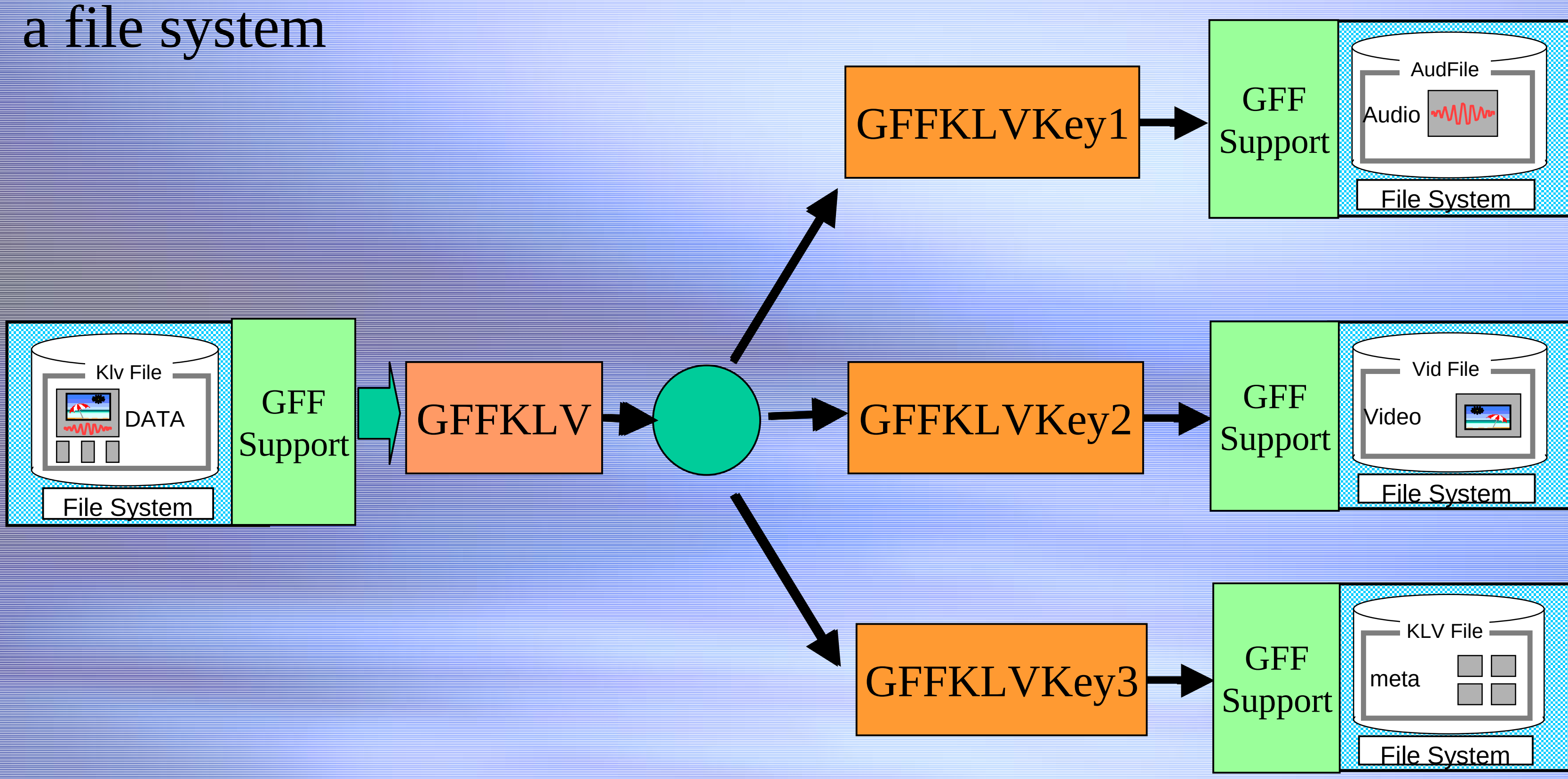


INESC PORTO

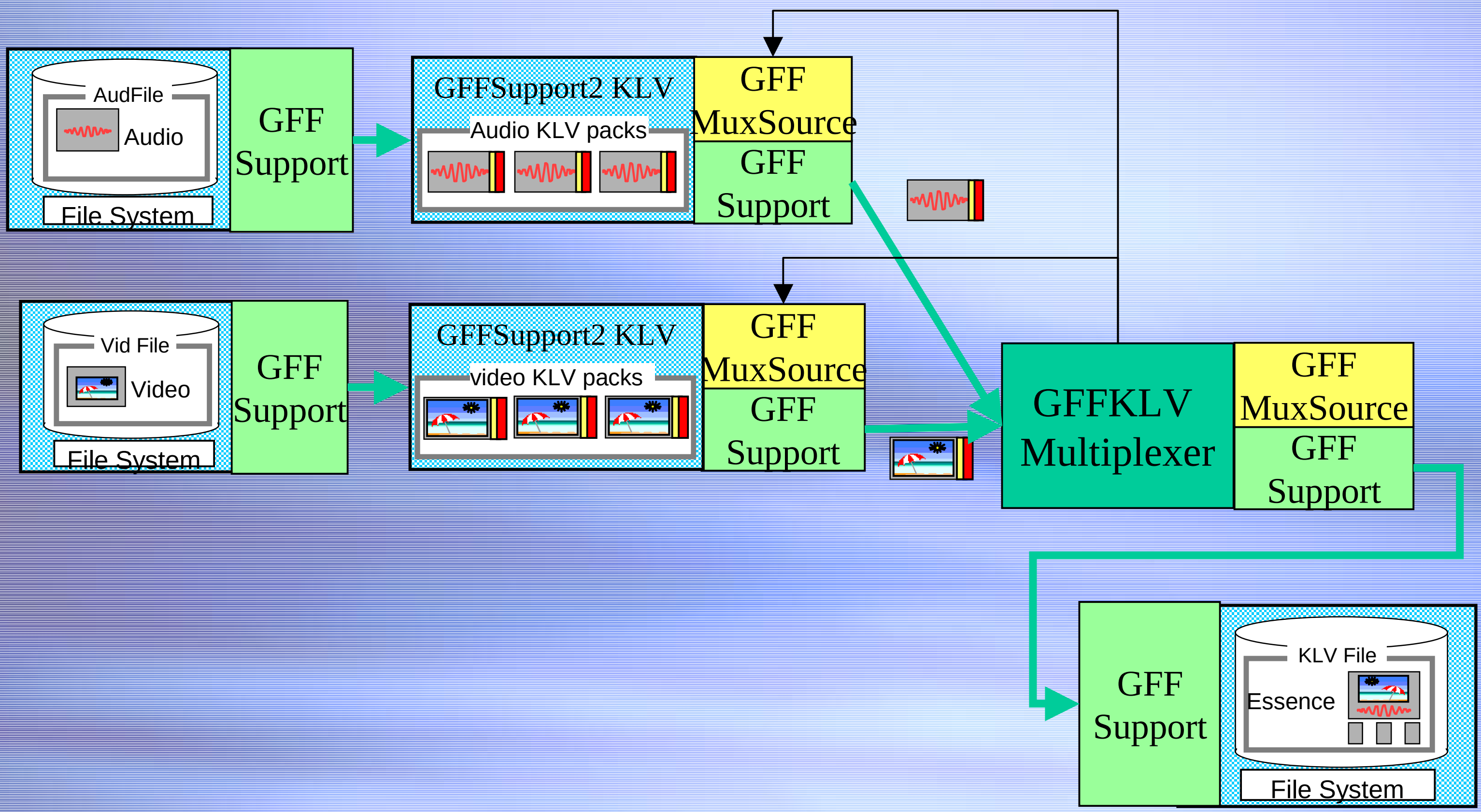
G-FORS

Example application (demux)

Application that uses a L2 (KLV layer) access to a file stored in a file system



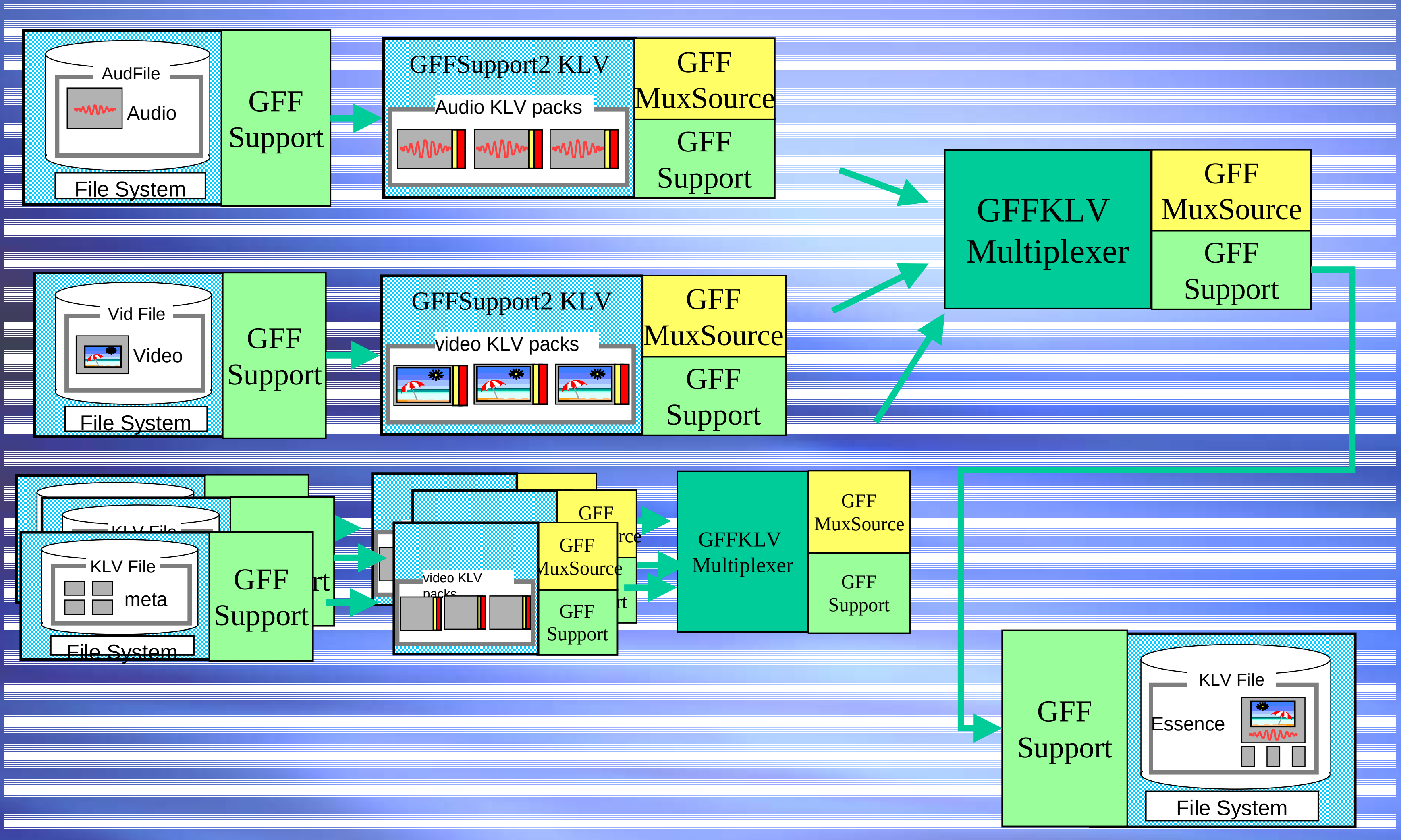
Simple KLV packetizer



INESC PORTO

G-FORS

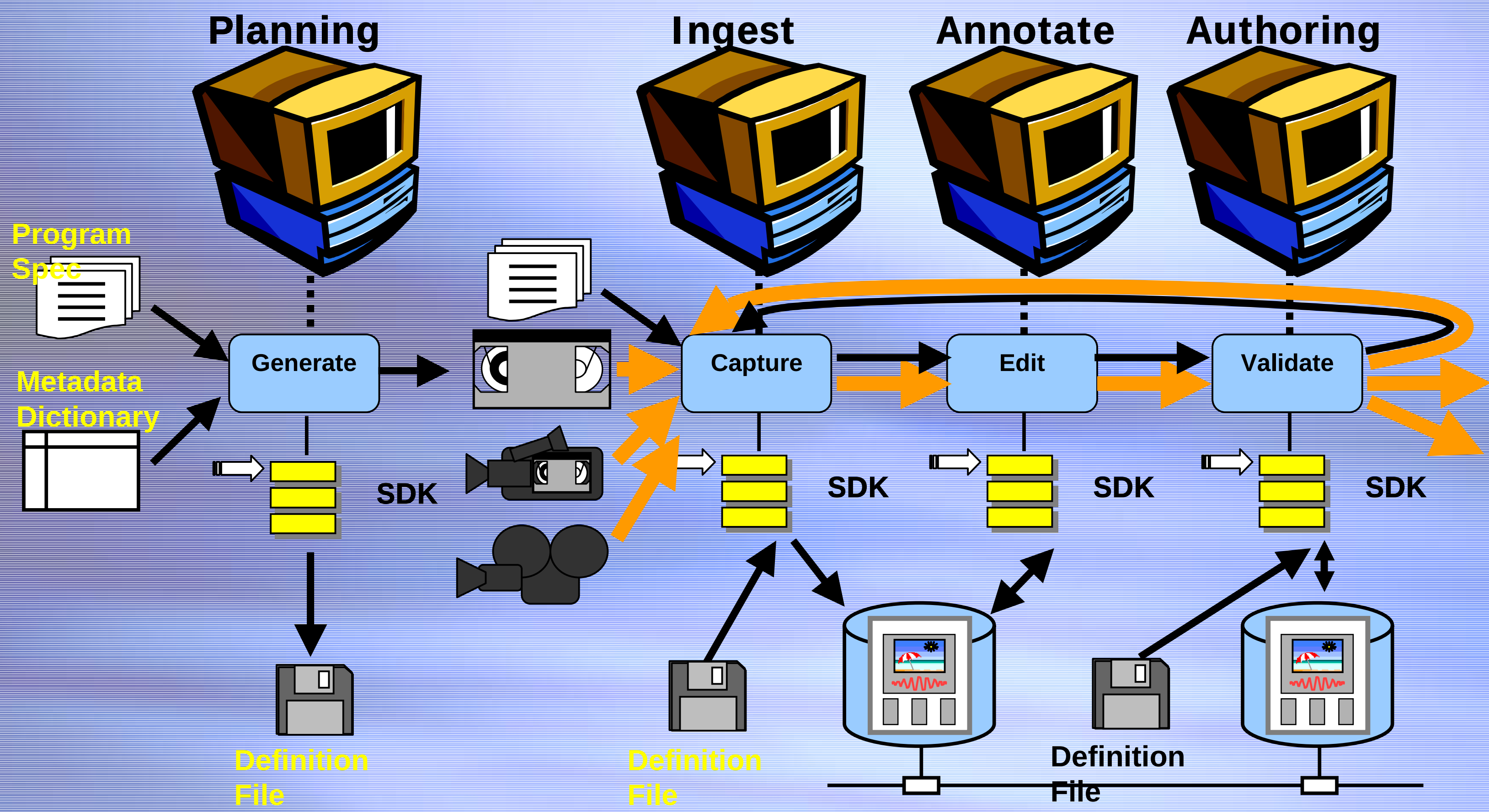
Multiplexing application



INESC PORTO

G-FORS

Program creation



INESC PORTO

G-FORS

Questions?

Places to look for more information:

www.g-fors.com

www.pro-mpeg.org

www.AAFassociation.org

An Optimal File Distribution Model for Data Broadcasting

Edwin A. Heredia

Microsoft Corporation
1065 La Avenida,
SVC-3 Mountain View, CA 94043
eheredia@microsoft.com

With the advent of terrestrial digital TV, the conventional 6 MHz RF band can be used to support not only traditional TV programs (audio and video) but also data broadcasting services. The advantage of such services will be the opportunity to cover large segments of the population with bit rates much higher than the typical telephone modems used nowadays. The disadvantage of data broadcasting using terrestrial TV transmission is its one-way-directionality. Data flows only in one direction from the server to the client while the existence of a reverse path is less likely. However, multimedia documents such as the electronic versions of newspapers and magazines are good candidates for unidirectional transmission since they may be designed without requiring interactivity with a server. This class of documents requires the delivery of a large collection of files or objects such as images, text, animations, sound, video clips, and others. In this presentation we will examine a method to deliver those objects in such a way that user access time is minimized.

In this presentation we examine the problem of file delivery using data carousels. Transmission using data carousels requires that each of the application files be sequentially and periodically emitted during a certain time segment that constitutes the service duration. From the decoder's perspective, the detection of files in carousels will trigger the download process. Depending on the amount of available storage, the decoder may choose to cache all the files prior to running the application or, instead, it may utilize a caching strategy of its own. From the emitter's perspective, it is necessary to transmit the collection of files in such a way that even decoders with little or no caching storage may be able to run the application. All data broadcasting standards indicate encapsulation formats that should be followed for data carousel transmission. However, little or no research has been done in the area of strategies to populate efficiently data carousels with files. If we have a large collection of files with heterogeneous properties such as size and access probabilities, is there a way to group them in one or more carousels for efficient transmission? This presentation addresses this problem and offers one strategy that attempts to minimize file access in non-caching decoders.

In this presentation we examine the use of multiple streams, each of which carries a separate carousel, as a means to address the problem of efficient delivery of files with heterogeneous size and access probabilities. We show that the multiple-stream strategy can be used in such a way that file access time is minimized and therefore, for a given broadcast bandwidth, the proposed file distribution model gives an optimal arrangement. Furthermore, by using the same algorithm repeatedly, one can determine the minimal required bandwidth for a given access time.

An Optimal File Distribution Model for Data Broadcasting

Edwin A. Heredia

Microsoft Corporation Silicon
Valley Campus, Bldg. 3
1065 La Avenida,
Mountain View, CA 94043
eheredia@microsoft.com

ABSTRACT

With the advent of terrestrial digital TV, the conventional 6 MHz RF band of NTSC systems can be used to support not only traditional TV programs (audio and video) but also data broadcasting services. The advantage of such services will be the opportunity to cover large segments of the population with bit rates much higher than the typical telephone modems used nowadays. The disadvantage of data broadcasting using terrestrial TV transmission is its one-way-directionality. Data flows only in one direction from the server to the client while the existence of a reverse path is less likely. However, multimedia documents such as the electronic versions of newspapers and magazines are good candidates for unidirectional transmission since they may be designed without requiring interactivity with a server. This class of documents requires the delivery of a large collection of files or objects such as images, text, animations, sound, video clips, and others. Standards have been proposed to establish data carousels for data delivery. However, these standards only indicate encapsulation formats required for data identification, signaling, and transmission. One problem that remains unsolved from the perspective of the data server is how to organize a large collection of files in carousels. In this paper we will examine an optimal method to organize files with heterogeneous properties of size and access probabilities in such a way that user access time is minimized.

1. INTRODUCTION

Terrestrial Digital TV, as defined in the US by ATSC *A/53* [1], offers the option to deliver bit streams at rates as high as 19.3 Mbits/s in every 6 MHz channel of the original NTSC system. This bandwidth may be used in different manners. It may be used to offer 3 or 4 conventional TV services (audio and video) in standard definition, it may be used for one high-definition channel, or it may be used to deploy several data broadcasting services such as HTML enhancements or Java-based software. These data services may be independent of the TV programs, or they may be tied up with the TV programs themselves.

An important part for the development of multimedia transmission services is the standardization of data broadcasting. Both of the major players, DVB in Europe and ATSC in the United States, have agreed to use DSM-CC as the base encapsulation protocol for data carousel services. For transmission, files are segmented first into one or more DSM-CC blocks, and later each block is divided once more into multiple Transport Stream packets in accordance with the MPEG-2

Systems protocol. The resulting 188-byte packets are multiplexed with others sharing the same Transport Stream, and after channel encoding, they modulate carrier signals using 8- VSB (See Fig. 1).

While DSM-CC and other upper-layer data broadcasting protocols indicate how to modularize and identify files within the data streams embedded in the Transport Stream, they do not provide methods to efficiently distribute objects in the streams. If a small number of objects is transmitted per document, then such an organization may not be required. However, multimedia documents such as existing WWW newspapers and magazines are typically composed of thousands of files with different sizes and access requirements. For large collections of documents, some intelligent file distribution method over multiple streams becomes necessary to reduce file access delay and save bandwidth.

In this paper we address such a problem. Based on object sizes and their access probabilities, we develop a method to distribute objects in multiple streams in such a way that the average file access time is minimized. We show that the resulting optimization problem can be transformed into a particular form of the quadratic allocation model for which an algorithmic solution has been developed. With bandwidth being very likely the most important commodity in the determination of broadcast costs, methods as the one introduced here are needed for the efficient use of bandwidth in the deployment of future multimedia broadcast services.

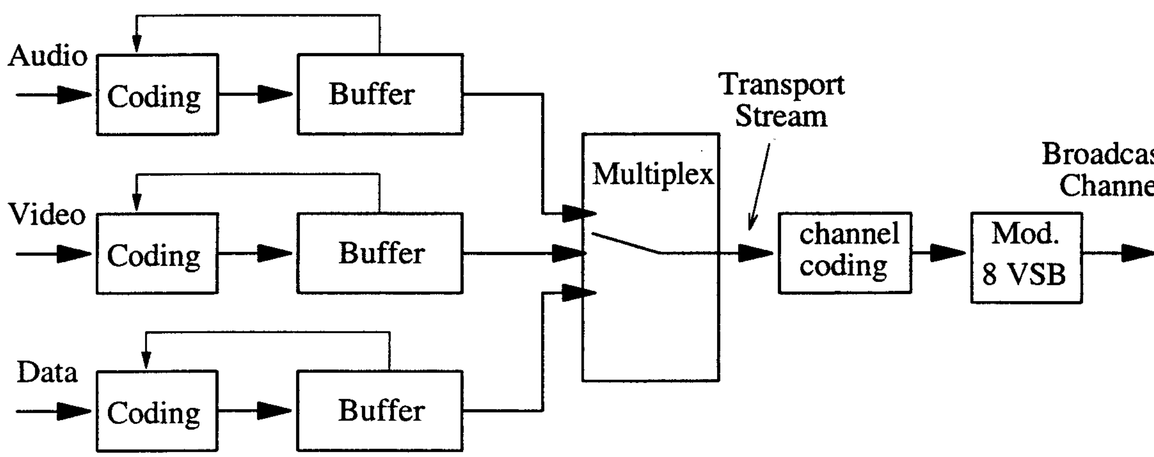


Figure 1. Encoding system architecture for Digital Television.

2. STREAMS FOR DATA BROADCASTING

The ATSC standard *A/53* [1] defines rules and constraints to build a Transport Stream based on the definitions of the MPEG-2 Systems protocol [5, 6]. The Transport Stream is the collection of 188-byte length packets delivered over a single 6 MHz frequency band. When using ATSC, the communication channel throughput is around 19.3 Mbits/s. This digital pipeline can itself be partitioned into multiple individual streams, each of which may use a guaranteed portion of the total bandwidth. Here we use the term *stream* to refer to the collection of packets identified by a unique packet identifier or PID and transmitted at approximately constant bit rate (CBR).

Figure 1 illustrates the data multiplexing process that leads to the Transport Stream. Buffers with occupancy feedback are used to control and guarantee the rate of individual streams. However, due to video traffic and multiplexing priorities, data packets can be dispersed when inserted in the Transport Stream. In practice, such dispersion modifies the actual bit rate moving it above or below the intended target, but the use of receiver buffers ensure that these fluctuations have no major effects.

A sequence of files transmitted one after the other and repeated periodically as part of a particular stream is called a data carousel. The periodic transmission of files allows users to access files randomly. Figure 2 shows a time-bandwidth diagram for a single stream showing the sequence of objects (or files) transmitted in the stream. In this example, two carousel groups distinguished by their colors are shown in the figure. The time segment occupied by the object represents the interval required for transmission of all the packets that compose the object. Notice that objects are transmitted sequentially with no free slots in between. Also in the same figure, the repetition time for object q_1 is explicitly indicated. A long repetition time for an individual object implies a long user access delay during downloading.

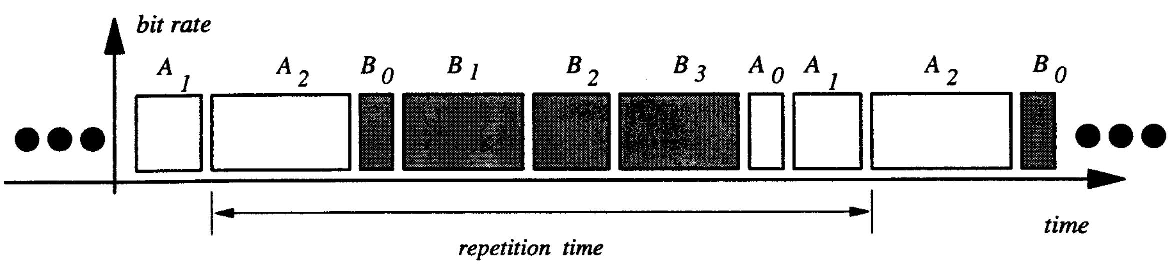


Figure 2. Access time components when retrieving objects from a broadcast stream.

For large documents with hundreds or thousands of files, avoiding long access delays may require the use of multiple streams. For example, high priority files should be placed in streams where small repetition times can be guaranteed, whereas low priority files could be queued together in separate streams. This is the problem addressed in this paper. Based on a priority measurement such as the object access probability, we develop an optimal allocation method to place objects in the proper streams in such a way that the average access time per object is minimized. The original results for this paper were described by the author in [9].

3. OBJECT ACCESS TIMES

A collection of objects (files, pages, or directories) Q = $\{q_1, q_2, \ldots, q_M\}$ is sequentially streamed at a constant bit rate of b bits per second as shown in Fig. 2. The access time for object k is defined as the time required for having the object available to any type of application software. Figure 2 illustrates that object access times have two components. The first one is the *wait time,*

w, from the instant when the object is requested to the instant when the object appears in the stream.
The second component is the *download time* and represents the time required to recover all the object packets from the stream. If S_k is the size in bits for object q_k and if b is the stream bandwidth, then the download time is S_k/b. Consequently, the access time for object q_k is given by

$$t_k = \frac{S_k}{b} + w \tag{1}$$

If the object request happens to be just before the object appearance in the stream, then the wait time will be null. However, if the request instant is slightly after the first object header bytes, then w will be maximum and equal to the time needed for the object to circulate and reappear in the stream. Figure 2 shows three examples of possible wait times when accessing object q_7.

Consequently, the wait time w is a random variable uniformly distributed between 0 and W_{max}, with

$$W_{\max} = \frac{\sum_{m=1}^{M} S_m}{b} \tag{2}$$

If $E\{\ \}$ denotes the expectation operator, then $E\{w\} = W_{max}/2$, and therefore:

$$E\{t_k\} = \frac{S_k}{b} + \frac{\sum_{m=1}^{M} S_m}{2b} \tag{3}$$

A WWW server with Internet access can register the number of times each page is accessed over a period of time. Based on this information, an empirical measure of access probabilities can be found for the web sites and by extension for the files that compose the pages. Let $p(k)$ be the access probability corresponding to the k-th object of the document object collection Q, then the overall average access time is given by:

$$t_A = \sum_{k=1}^{M} E\{t_k\} p(k) \tag{4}$$

For the single stream case described in the previous section, $E\{t_k\}$ is defined in Equation 3. In the next section, we compute $E\{t_k\}$ for the multi-stream case and use t_A to develop an optimization problem whose solution gives the object allocation for minimal access time.

4. DISTRIBUTIONS FOR MULTIPLE STREAMS

For documents with a small object collection, a single stream is enough to carry the entire collection. In fact, multiple documents may be streamed together through the use of data carousel grouping such as the one provided by DSM-CC structures. However, for large documents the delivery may require multiple streams to guarantee small access delays. Streams are recognized by their packet identifier (PID) that labels the MPEG-2 Transport Stream packets. Current technology enables stream tuning by PID filtering in transport processing chips. Stream tuning is fast and easy while the overhead comes from the wait and download times described in the previous section.

Audiovisual streams, program and system information (PSIP), and other MPEG-2 functions utilize large number of PIDs. Therefore, while multi-streaming is important for reducing access times, the number of defined streams should simultaneously be kept as small as possible due to hardware limitations in the number of PID filters that can be established at a given time.

Once more consider the set of M objects $Q = \{q_1, q_2, ..., q_M\}$ which for transmission purposes will be broadcast using multiple streams. Each of the objects has a size S_k and an access probability $p(k)$ for $k=l$, 2, ..., M. Let N represent the number of available streams and let C_j designate the j-th stream with bandwidth b_j. An example of the distribution of 11 objects over three streams with different bandwidths is illustrated in Fig. 3.

An assignment matrix $X = [x_{ij}]$ of size M x N is defined here, whose elements indicate whether an object belongs or not to a certain stream, that is

$$x_{ij} = \begin{cases} 1 & q_i \in c_j \\ 1 & q_i \notin c_j \end{cases} \qquad (5)$$

Assuming in principle, that object q_k belongs to the arbitrary streaming channel C_j, then equation 3 can be invoked to compute $E\{t_k\}$. This gives

$$E\{t_k\} = \frac{S_k}{b_j} + \frac{\sum_{i=1}^{M} S_i x_{ij}}{2b_j} \qquad \textit{if qk } \mathrm{E\ C}\ j \qquad (6)$$

The assignment matrix can be used once more to remove the "if' clause of the previous expression, which gives

$$E\{t_k\} = \sum_{j=1}^{N} \left[\frac{S_k}{b_j} + \frac{\sum_{i=1}^{M} S_i x_{ij}}{2b_j} \right] x_{kj} \qquad (7)$$

The assignment matrix X is precisely the term we would like to determine following some type of optimization method.

Substituting equation 7 into 4, after some algebraic manipulation, it is possible to demonstrate that the overall average access time for a multi-stream object allocation, T_A, can be written as

$$T_A = \sum_{k=1}^{M}\sum_{j=1}^{N}\alpha_{kj}x_{kj} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{M}\beta_{ijk}x_{kj}x_{ij} \qquad (8)$$

where the equation constants are defined as

$$\alpha_{kj} = \frac{p(k)S_k}{b_j} \quad , \qquad \beta_{ijk} = \frac{p(k)S_i}{b_j} \qquad (9)$$

Therefore, the optimization problem can be stated as follows.

Find the assignment matrix X = [x_{ij}] *such that the overall average access* T_A is minimized, *subject to the following two constraints:*

1. $x_{ij} \in \{0,1\}$ for all values of i and j
2. $\sum_{j=1}^{N} x_{ij} = 1$

The first constraint implies that the solution space is binary, meaning that an object mayor may not belong to a certain stream, whereas the second constraint indicates that an arbitrary object can be assigned to one and only one stream. Because of the quadratic form of the cost functional T_A described in equation (8) and because the solution space is binary, this optimal allocation problem can be classified as non-linear integer programming problem of the zero-one kind.

In conventional data broadcasting applications, the stream bandwidths are negotiated a priori, and once the broadcast server admits such bandwidths in guaranteed mode, the rates are maintained at the defined levels. When all the selected data broadcast streams have the same bandwidth, the optimization problem can be further simplified. In this case, after defining the terms

$$C = \frac{1}{b}\sum_{k=1}^{M} S_k\, p(k) \quad , \qquad a_{ik} = S_i\, p(k) \qquad (10)$$

then, the overall average access time becomes

$$T_A = C + \frac{1}{2b}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{M} a_{ik}x_{kj}x_{ij} \qquad (11)$$

From their definitions, it is clear that C and b are positive numbers, consequently, for the equal bandwidth case, a simplified cost function results:

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{M} a_{ik}x_{kj}x_{ij} \qquad (12)$$

subject to the same constraints as the previous non-uniform bandwidth case, that is: $x_{ij} \in \{0,1\}$ and $\sum_{j=1}^{N} x_{ij} = 1$.

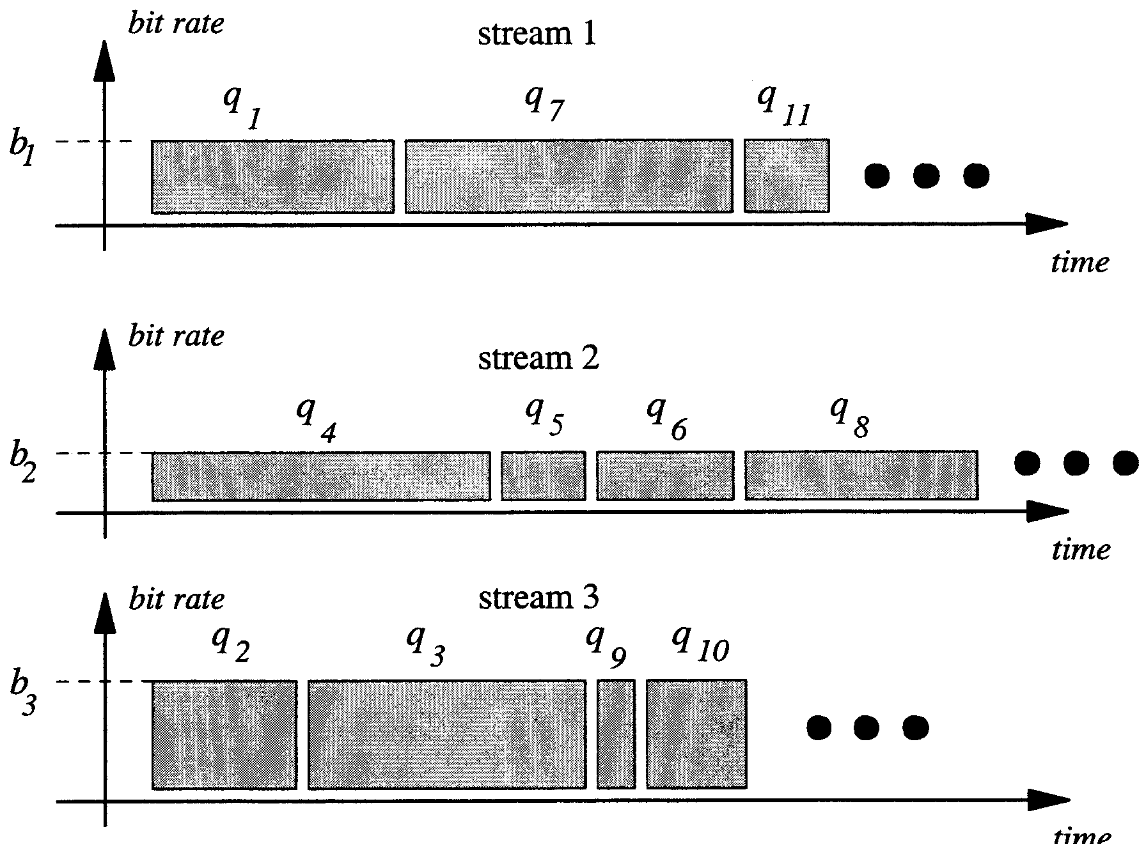


Figure 3. An example of object distribution over multiple streams.

5. OPTIMIZATION ALGORITHM

The optimization model defined by equation 12 is similar to a particular form of the generalized quadratic assignment problem. This form is normally obtained when studying classroom scheduling problems (CSP) [8] or the allocation of interacting activities to facilities [2, 3, 4]. Like most of the known quadratic assignment problems, the existence of nonlinear interaction terms makes these, otherwise simple problems, NP hard.

Carlson and Nemhauser have proposed an optimization algorithm for the CSP problem applicable when the coefficient matrix is symmetric with null diagonal [2]. Under these conditions, local minima can be found through a recursive process. However, for the stream allocation case, it is evident from equation (10) that the coefficient matrix A is not symmetric and has, in general, a non-null diagonal. We show next that a reformulation of the problem is possible to meet the constraints imposed by Carlson and Nemhauser.

Let $y_{ik} = \sum_{j=1}^{N} x_{ij} x_{kj}$. By inspection, it is evident that the matrix *[Y*ik*]* is: (1) symmetric, (2) has ones as its diagonal elements, and (3) has only ones or zeros as elements. Consequently, using these properties, equation 12 can be re-written in terms of Y_{ik} as:

$$J = \sum_{i=1}^{M} S_i p(i) \quad + \quad \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{M} \tilde{a}_{ik} y_{ik} \qquad (13)$$

where

$$\tilde{a}_{ik} = \begin{cases} (a_{ik} + a_{ki})/2 & k \neq i \\ 0 & k = i \end{cases} \qquad (14)$$

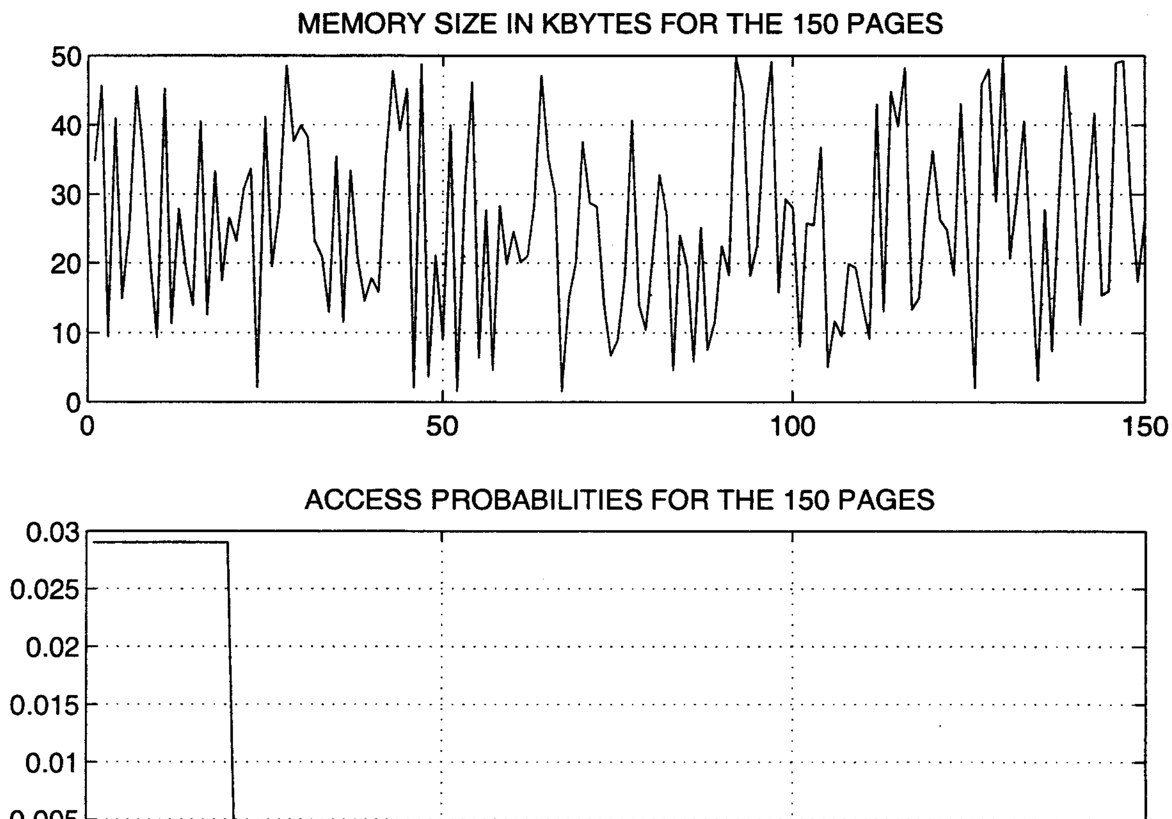


Figure 4. The top diagram shows the distribution of file sizes for each of the 150 files. The bottom diagram shows the distribution of access probabilities.

The first summation of equation 13 is a constant and therefore, the optimization problem can be restated in simpler terms as minimize the function z which is defined as

$$z = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{M} \tilde{a}_{ik} x_{ij} x_{kj} \qquad (15)$$

subject to the same constraints as before. The matrix $\tilde{A} = [\tilde{a}_{ik}]$ is symmetric with null diagonal, and therefore satisfies the conditions required for using the Carlson-Nemhauser algorithm. This algorithm was implemented as described in reference [2]. The only difference in our implementation of the algorithm is that we used two starting feasible solutions to calculate two optimal allocation matrices. Because the algorithm gives local minima, then we chose the best of the two solutions as the adopted minimum.

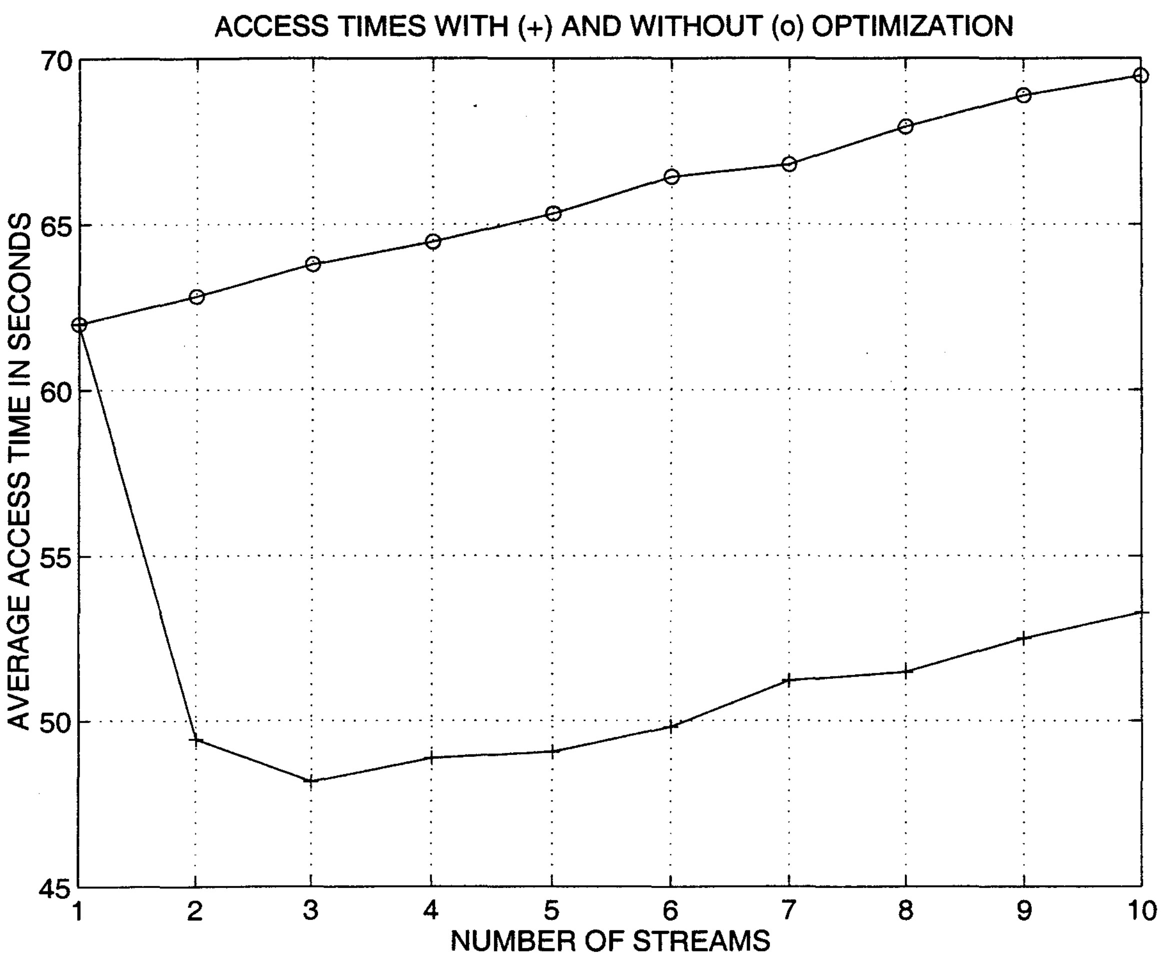


Figure 5. Average user access times as a function of the number of streams for blind assignment (0) and optimal allocation (+).

6. APPLICATION EXAMPLE

Consider the problem of finding the proper number of streams to use when broadcasting a large WWW document. For this example, we assume that the document is composed of 150 pages with sizes as shown in Figure 4. The sizes were obtained using a uniform random number generator between 1 and 50 Kbytes. Using these values, the total document size is about 3.8 MB. For a probability distribution, we assume that from the total of 150 pages, 20 are considered highly likely to be accessed (hot pages) while the remaining 130 have low access probabilities (see Fig. 4). Different probability and size distributions have no impact on the method, since the optimization procedure is carried out without any assumptions regarding these distributions.

Assuming an available broadcast bandwidth of 250 Kbits/s, we want to determine an efficient way to use the bandwidth for over-the-air broadcast. One option is to allocate the. pages uniformly (page 1 placed in stream 1, page 2 to stream 2, and soon). The second option uses the optimization procedure described in this paper. Figure 5 shows the results when the process is repeated for different cases with the number of streams ranging from 1 to 10. The figure compares the average access time for each of the cases.

When one stream is used the available bandwidth is entirely dedicated to that single stream. When *N* streams are used, each receives its corresponding fraction of bandwidth. It is evident in this case that blind multi-streaming (that is, without using optimization) produces no benefits and should be, in general, avoided. When using optimization, the situation changes drastically. Figure 5 shows that multi-streaming helps reducing the average access time from 62 (one stream) to 48 seconds (three streams). Without optimization, such a reduction in access time can only be accomplished by increasing the available bandwidth from 250 Kbits/s to 320 Kbits/s (almost 30% of bandwidth increase!) Since bandwidth is likely to become a costly commodity in multimedia broadcast services, optimization methods like the one presented in the paper are required to maximize efficiency.

7. CONCLUSIONS

Data broadcasting protocols such as the ones developed by ATSC [10] and DVB [11] will be used for the delivery of large multimedia documents composed of hundreds or thousands of files. The problem of distributing a large collection of multimedia files among multiple broadcast streams is studied in this paper. Based on measurement of priority (the object access probability), we demonstrate that the file allocation problem can be classified as a quadratic optimization problem of the zero-one kind. We show that through algebraic manipulation of the problem constants, the cost functional may be re-written in a form that is compatible with a similar problem studied by Carlson and Nemhauser [2]. The parameter that gets minimized is the average user access time when downloading any page of the collection. We show as an example, that for a typical multimedia document composed of 150 pages with sizes ranging from 1 KB to 50 KB, the optimization method reduces the access delay from 62 seconds (one stream) to 48 seconds (three streams). A reduction that otherwise would be achieved by increasing the transmission bandwidth about 30%. Bandwidth reduction techniques, like the one described in the paper, will be necessary once multiple data services and television programs share the same communication channel and compete for bandwidth.

8. REFERENCES

1. ATSC *A/53, Digital Television Standard*, 1995

2. R. C. Carlson and G. L. Nemhauser, "Scheduling to Minimize Interaction Cost", *Operations Research,* Vol. 14, No.1, pages 52-58, Jan -Feb 1966.

3. H. Greenberg, "A Quadratic Assignment Problem without Column Constraints", *Naval Research Logistics Quarterly,* Vol. 16, No.3, pages 417-421, Sept. 1969.

4. H. Greenberg, *Integer Programming,* Academic Press, 1971

5. B. G. Haskell, A. Puri, and A. Netravali, *Digital Video: An Introduction to MPEG-2,* Chapman and Hall, 1997.

6. ISO 13818-1, Coding of Moving Pictures and Associated Audio -Part 1: Systems, 1996.

7. ISO 13818-6, Coding of Moving Pictures and Associated Audio -Part 6: Extensions for DSM-CC, 1996.

8. M. Padberg, and M. Rijal, *Location, Scheduling, Design and Integer Programming,* Kluwer Academic, 1996.

9. E. A. Heredia, "Optimal Object Allocation for Multimedia Broadcast", *Proceedings IEEE International Conference on ASSP,* Seattle, W A, 1998.

10. ATSC *A/90, Data Broadcast Standard,* 2000

11. ETSIEN 301192, DVB *specification for data broadcasting,* 1997

Data Broadcasting Standards Overview

Michael A. Dolan

Industry Consultant,
miked @tbt.com

Data broadcasting covers many technical areas. This presentation provides a brief overview of the application requirements for the entire system, including authoring through the receiver. Then, a review of all the public standards efforts is provided in this context, including ARIB, ATSC, DVB, and SMPTE. Finally, a few comments are provided on efforts at harmonization of these efforts. An extensive references list is included which will supplement the presentation for further reading in the field.

NIST/ATSC Symposium: End-to-End Data Services, Interoperability and Applications

Data Broadcasting Standards Overview

Michael A. Dolan

20-June-2001

Overview

- Data Application Scenarios
- Data Application Infrastructure
- General Technologies
- Standards Organizations
- Standards Work
- Example
- Detailed References

Data Application Scenarios

- Captioning/Teletext (for 25+ years!)
- IP Transport
- Webcasting/Datacasting
- “Buy-This-Now”
- “Send Me More Information”
- Polling
- Real Time Gaming
- Web Browsing

Captioning & Subtitling

Image Courtesy of MRG Systems

Teletext

  Image Courtesy of MRG Systems

Captioning (US)

- Analog TV Closed Captioning (ATVCC) Specification
- Regulatory
- Used mainly for closed captioning
- Includes Extended Data Services (XDS)
  - National Weather Service (NWS) Alerts
  - Misc Service and Program Metadata

Teletext (EU*)

* term used loosely to include UK, etc

- ITU Recommendation
- Widespread deployment in EU
- Never really caught on in the US
- Carousel of pages
- Applications such as train schedules
- Often interactive (i.e. backchannel comm.)

IP Transport

- Support for MPEG as a network UDLR*
- Initial development being done with this
- Well understood from Internet and VBI
- Basis for proprietary multicast webcasting & datacasting products and services
- May be useful for limited unicast forward channel, but doesn't scale well

* UniDirectional Link Route

One-Button Interaction

- "Buy-This-Now"
  - pre-arranged business model and shipping info
- "Send Me More Information"
  - pre-arranged mailing info
- Polling
  - Yes/No/Maybe

Gaming

- Real-time interaction
- Play against TV or against other viewers
- Game shows in use now in the US
  - Jeopardy®
  - Wheel of Fortune®
  - Who Wants to be a Millionaire®

Web Browsing

- “Get More Information” in real time
- Links to web sites from broadcast programs
- Business model hard to support when distracted from the broadcast
- PVR Technology may make the business model work for this application

The Application Infrastructure

- Data Applications need:
  - Basic Transport (i.e. MPEG)
  - Video/Audio Framework with Metadata
  - Data Framework with Metadata
  - Authoring Application Environment
  - Receiver Application Environment

Data & Timing Models

- Data Models
  - Files
  - Streams
  - IP Packets
- Timing Models
  - Asynchronous
  - Synchronized
  - Synchronous

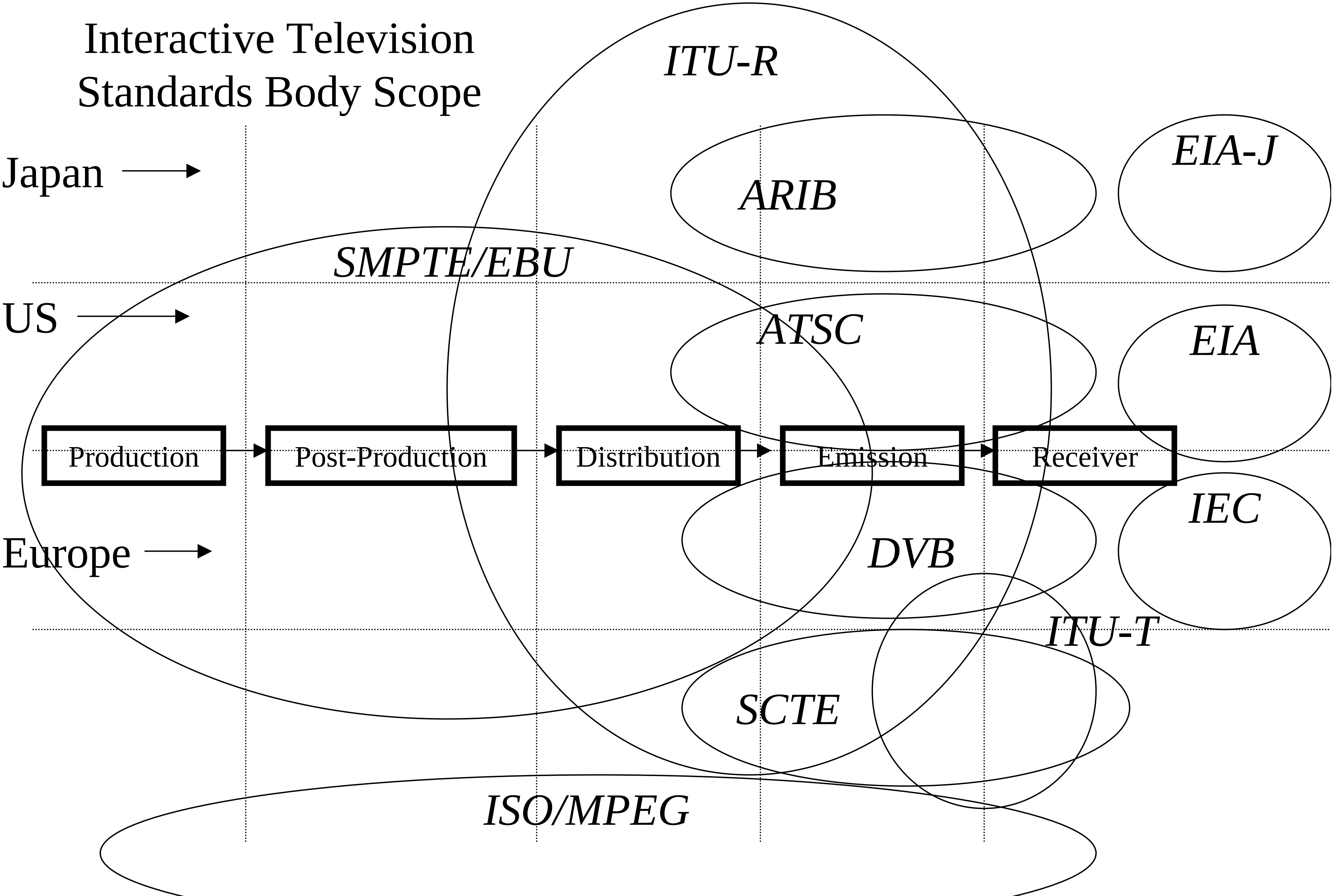
General Environment Technologies

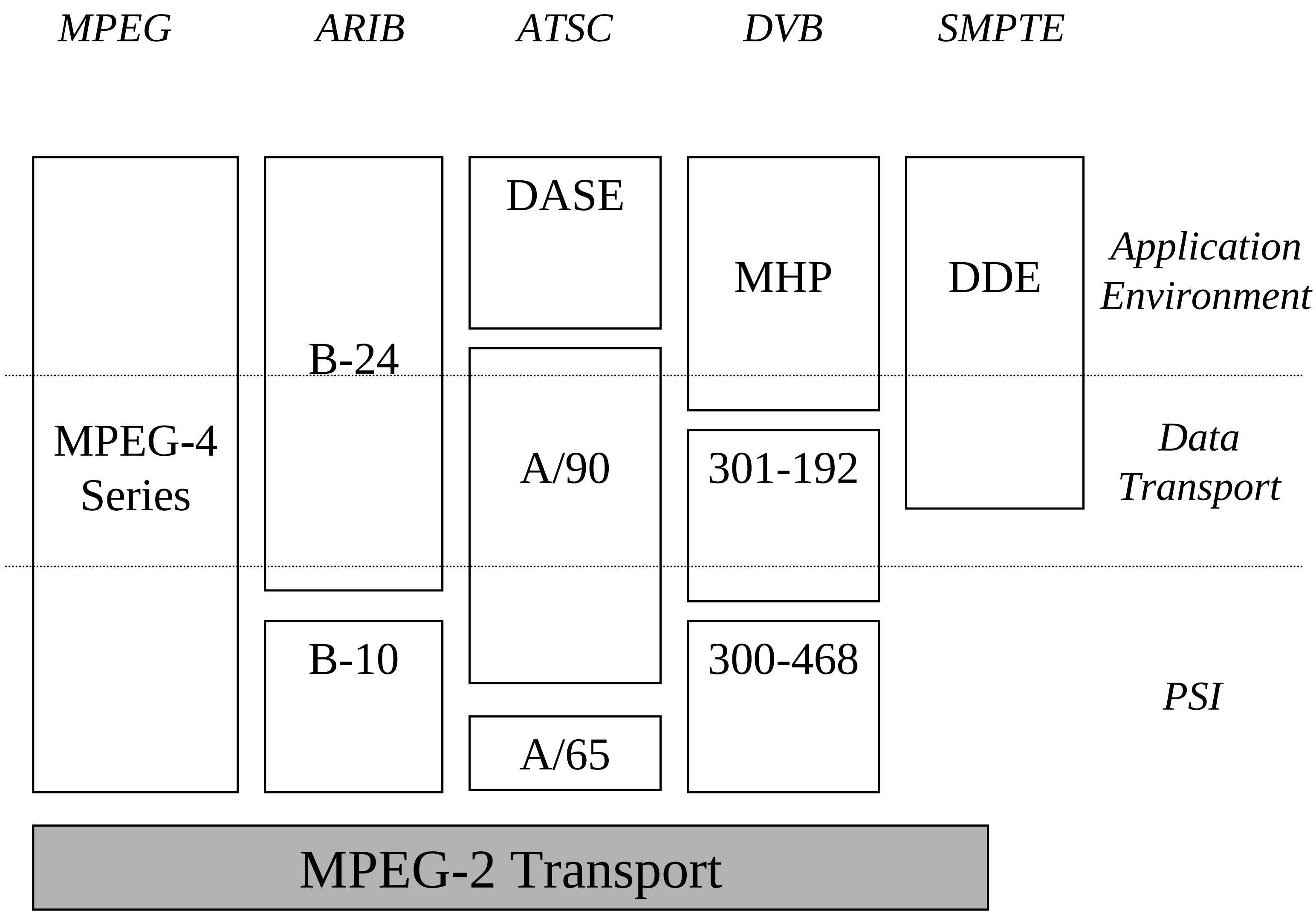
- ISO MPEG-2, DSM-CC, IETF IP Multicast
- HTML/XHTML
  – W3C Recommendations "the web"
- Java®
  – Sun Microsystems®
- Proprietary Systems
  – Canal+®, OpenTV®, Wink®

Standards Organizations

- Studio, Facility & Distribution
  - EBU, ISO/MPEG, ITU, SMPTE
- Digital Transports
  - ARIB (JP), ATSC (US), ETSI/DVB (EU)
  - ISO/MPEG (INT), SCTE/OCAP (US)
- Consumer Electronics
  - EIA (US), IEC (EU), EIA-J (JP)

Interactive Television Standards Body Scope





MPEG
ARIB
ATSC
DVB
SMPTE
DASE
MHP
DDE
Application Environment
B-24
MPEG-4 Series
A/90
301-192
Data Transport
B-10
300-468
PSI
A/65
MPEG-2 Transport

20-June-01

ISO/MPEG-4 version 2

- MPEG-2 & MPEG-4 Transports
- Data & Timing Models
  - MPEG-4 Encapsulations (Files & Streams)
  - Synchronized
- Focus on Video with (File) Object Composition
- MPEG-J
  - JVM, API's still tbd

ARIB B24*

* no public information on work in process

- MPEG-2 Transport
- Data & Timing Models
  - Data Carousel (Files), Streams, IP Packets, Triggers
  - Asynchronous, Synchronized
- BML (XHTML-derivation)
  - CSS1(+), DOM1(+), ECMAScript

ATSC DASE 1.0*

* includes public work in process

- MPEG-2 Transport
- Data & Timing Models
  - Data Carousel (Files), Streams, IP Packets, Triggers
  - Asynchronous
- JVM
  - PJAE, Java TV, org.atsc.*
- XDML
  - CSS2(-), DOM2(-), ECMAScript

DVB MHP 1.x*

* includes public work in process

- MPEG-2 Transport
- Data & Timing Models
  - Object Carousel (Files), Streams, IP Packets, Triggers
  - Asynchronous, Synchronized
- JVM
  - PJAE, Java TV, org.dvb.*
- XHTML
  - CSS2(-), DOM2(-), ECMAScript

SMPTE DDE-1*

* based on public ATVEF spec

- Transport-Independent
- Data & Timing Models
  – UHTTP (Files) carried in IP Multicast, Triggers
  – Asynchronous
- HTML-4
  – DOM0, CSS1, ECMAScript

Comparison Matrix

Technology	ISO/MPEG	ARIB B24	ATSC DASE	DVB MHP	SMPTE DDE
Analog Transport					X
MPEG-2 Transport	X	X	X	X	[1]
MPEG-4 Transport	X				
Files	X	X	X	X	X
Streams	X	X	X	X	
IP Packets		X	X	X	
Triggers		X	X	X	X
JVM	X		X	X	
PJAE	[2]		X	X	
JavaTV			X	X	
HAVi			X	X	
org.atsc.*			X		
org.dvb.*				X	
HTML/XHTML	X	X	X	X	X
DOM		X	X	X	X
CSS		X	X	X	X

[1] Transport Independent via IP

[2] MPEG-J is early work in process

The $64 Question

- **What is really needed for “interactive television” and “data broadcasting”?**
- There are arguments for both extremes
  - None - program director fully controls the viewer experience, so it can simply be done in the video
  - Full edge-of-the-seat web-like interaction
- Answer: Something in between, but first...

Example Trivial Application - IP

- *Network* is the MPEG Program (maybe)
  - the channel, service, or virtual channel
- MPE using DSMCC Addressable Sections
  - small variations between systems
- Signaled in PMT and special tables
  - somewhat wider variation between systems
- Announcement in Event tables
  - virtually no compatibility between systems

Impedance Mismatches Abound

- Studio - Distribution - Emission Interfaces
- Between Transport Systems
  - even the most trivial scenario can’t be easily done
- Between authoring environments
- Between receiver environments

Harmonization Efforts

- International scope required
- Multiple interface points in distribution
- Focus on emission/receiver:
  - ITU-R WP6M
  - ITU-T SG9
  - ITU JRG-1

References (Organizations)

- Association of Radio Industries and Business (**ARIB**), http://www.arib.or.jp
- Advanced Television Systems Committee (**ATSC**), http://www.atsc.org
- Digital Video Broadcast (**DVB**), http://www.dvb.org
- Electronic Industries Alliance (**EIA**), http://www.eia.org
- European Broadcast Union (**EBU**), http://www.ebu.ch
- European Telecommunications Standards Institute (**ETSI**), http://www.etsi.org
- International Telecommunication Union (**ITU**), http://www.itu.ch
- Moving Picture Experts Group (**MPEG**), ISO JTC1/SC29 WG11, http://www.cselt.it/mpeg
- Society of Cable Telecommunications Engineers (**SCTE**), http://www.scte.org
- Society of Motion Picture and Television Engineers (**SMPTE**), http://www.smpte.org

References (Transports)

- ARIB STD-B10, "*Service Information for Digital Broadcasting System*"
- ARIB STD-B24, “*Data Coding and Transmission Specification for Digital Broadcasting*”
- ATSC Standard A/65-A, "*Program and System Information Protocol for Terrestrial Broadcast and Cable*"
- ATSC Standard A/90, “*Data Broadcast Standard*”
- ETSI EN 300 468-2000, “*Specification for Service Information (SI) in DVB systems*”
- ETSI EN 301 192-1999, “*DVB specification for data broadcasting*”
- ETSI TS 101 812, “*DVB Multimedia Home Platform (MHP) Specification 1.0*”
- ISO/IEC 13818-1, “*Generic coding of moving pictures and associated audio information: systems*”
- ISO/IEC 13818-6, “*Digital Storage Media – Command and Control*”
- SCTE **draft** DVS 311, “*IP Multicast for Digital MPEG Networks*”

References (Content)

- ARIB STD-B24, “*Data Coding and Transmission Specification for Digital Broadcasting*”
- ATSC **draft** T3/S17 “*DTV Application Software Environment (DASE)”*
- EIA 608-B, “*Line 21 Data Services*” [ATVCC]
- EIA 708-B, “*Digital Television Closed Captioning* “ [DTVCC]
- ISO/IEC 14496-x, “*Coding of audio-visual objects*” [MPEG-4]
- ITU-R Recommendation BT-653-2, "*System B, 625/50 television systems*” [Teletext]
- SMPTE **Proposed** 363M, “*Declarative Data Essence, Content Level 1*”
- SMPTE **Proposed** 357M, “*Declarative Data Essence, IP Multicast Encapsulation*”

References (Other)

- Addison Wesley, Lindholm & Yellen, "*Java Virtual Machine Specification, Version 1.0*", ISBN 0-201-43294-3 [JVM]
- ECMA, "*ECMA-262 ECMAScript Language Specification*" [ISO 16262]
- McGraw-Hill, Chernock et al, "*Data Broadcasting: Understanding the ATSC Data Broadcast Standard*"
- Sun Microsystems, "*Java TV API Specification*" [JavaTV]
- Sun Microsystems, "*Java Media Framework, Version 1.0*" [JMF]
- Sun Microsystems, "*Personal Java Application Environment Specification, Version 1.2A*" [PJAE]
- W3C Recommendation, "*Document Object Model (DOM) Level 2 Core*"
- W3C Recommendation, "*Extensible Markup Language (XML) 1.0*"
- W3C Recommendation, "*HTML 4.01 Specification*"
- W3C Recommendation, "*XHTML 1.0: The Extensible HyperText Markup Language*"

Author Contact Information

- Michael A Dolan
  - PO Box 1673, Alpine, CA 91903
  - 619-445-9070
  - miked@tbt.com

- Acknowledgements
  - DIRECTV for their general support in the field
  - SMPTE for ongoing support in converging data broadcast standards
  - MRG Systems for several images www.mrgsystems.co.uk

DVB-MHP: Technical Overview and Commercial Status

Herve Creff

GM, Product Marketing
CANAL+ TECHNOLOGIES
www .canalplus-technologies.com
hcreff@canal-plus.fr

This presentation provides a technical overview of the DVB-MHP (Digital Video Broadcast project -Multimedia Home Platform) standard, delivers the schedule of the MHP initiative and discusses the commercial success of MHP.

MHP has been designed as a standardized, end-to-end and interoperable solution for digital interactive TV. The main goals are to ease the creation of interactive TV content and to allow for the emergence of a retail market for digital TV terminals.

At the client side, MHP is giving support to two software platforms, one (called DVB-J) being JavaTM-based, the other (called DVB-HTML) being Markup Language (ML)-based. From the system point of view, MHP is defining (for both DVB-J and DVB-HTML) an application model, a graphic model, a security model, interactive and broadcast protocols, a signaling protocol as well as a selection of content formats. MHP specifies three application profiles: Enhanced TV featuring local interactivity, Interactive TV featuring interactivity over the return path and Internet Access.

Frozen in February 2000 by DVB, MHP 1.0 has been published by ETSI in August 2000. MHP 1.1 (introducing DVB-HTML) is expected to be frozen by DVB in June this year. The MHP test suite should be adopted by DVB Q4, 2001.

The first MHP compliant terminals will appear on the retail market before the end of the year. MHP introduction is likely to be driven by the retail market and the deployment of Digital Terrestrial Television (DTT). The European Nordic countries (for all networks), Singapore (for DTT), Australia (for DTT) and Korea (for DTS) have officially announced their support to MHP . The issues that can potentially slow down the MHP penetration are the migration issue (from a legacy platform to MHP), the cost of an MHP terminal (wrt a legacy terminal) and the interoperability issue.

MHP is a solid technical solution for iTV that has to be transformed into a commercial success. 2002 should be the year of MHP .

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

DVB-MHP : technical overview, commercial status

DASE Symposium
20th of June 2001

Hervé Creff
GM, Product Marketing
CANAL+ TECHNOLOGIES
hcreff@canal-plus.fr

Mission and goal

- Mission : define a standardized, end-to-end and interoperable system for interactive digital TV
- Goal : ensure the development of an open horizontal market for digital TV terminals
- Goal : ease the creation of interactive TV contents

CANAL+ TECHNOLOGIES

The DVB-MHP specification

Technical overview

Schedule

Commercial status

Conclusion

- Supports two software platforms
  - DVB-J : a Java™-based platform comprising a Virtual Machine and a collection of Application Programming Interfaces (APIs), some specifically designed for the TV environment
  - DVB-HTML : a Markup Language (ML)-based platform comprising a TV-specific ML and a scripting language
- Defines for both DVB-J and DVB-HTML
  - An application model (application definition, application lifecycle, application management, resources management)
  - A graphic model
  - A signalling protocol
  - Transport protocols (broadcast and interactive)
  - Content formats for text, images, audio clips, fonts, etc.
  - A security scheme to authenticate applications and secure return channel transactions

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

The DVB-J packages

Technical overview

Schedule

Commercial status

Conclusion

- JDK APIs
  - java.lang (subset)
  - java.lang.reflect
  - java.io
  - java.util
  - java.util.zip (subset)
  - java.net (subset)
  - java.awt
  - java.beans
  - java.math.BigInteger
  - java.rmi (subset)
- Sun APIs
  - JavaTV (without javax.tv.carousel)
  - JMF 1.0
  - java.security, java.security.cert, JSSE (subset)
- HAVi APIs
  - awt extensions
  - TV widget set
- DAVIC 1.4.1 APIs
  - MPEG APIs (incl. Section Filtering API)
  - Tuning API
  - Resource Framework API
  - CA API
  - Locator API
  - JMF extensions
- DVB APIs
  - org.dvb.lang
  - org.dvb.event
  - DVB SI API
  - User Preferences API
  - DSMCC API
  - UI API (extended graphics)
  - Return Channel API
  - Application Management API
  - Persistent Storage API
  - CA permission API

CANAL+ TECHNOLOGIES

DVB-HTML (1/2)

- Fully in line with W3C recommendations: based on XHTML
  - Shares common basis with the XML world
- Defines
  - A content format (as a selection from XHTML 1.0 modularization)
  - A presentation format (as a subset of CSS2)
  - An event model allowing for a fine grained synchronization with A/V content
  - An interface to the document (as a subset of DOM2)
  - A script language (ECMAScript)

DVB-HTML (2/2)

- = HTML functionality + :
  - Complete integration within DVB-MHP standard (part of DVB-MHP 1.1)
  - Benefits from DVB-MHP application model
  - Benefits from DVB-MHP security scheme
  - Integration with DVB-MHP graphic model
  - Access to digital TV terminal resources through DVB-J APIs
  - Fine grained A/V synchronization
  - Remote Control type of navigation
  - Associated conformance regime to achieve interoperability

CANAL+ TECHNOLOGIES

Three application profiles

- Enhanced TV
  - Featuring local interactivity
  - TV browsing
  - Information retrieval services (news, sports, weather forecast, financial information, …)
- Interactive TV
  - Featuring interactivity over the return path
  - Allow for transactional applications : t-commerce
- Internet access

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

What’s new in MHP 1.1 ?

- DVB-HTML
- New APIs
  - Plug-in API
  - Bank card API (Open Card Framework subset)
  - Application storage API
  - Internet applications management API
- Application management over the return path
- Resident applications management
- Internet Access profile

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

Schedule

Technical overview

Schedule

Commercial status

Conclusion

- MHP 1.0
  - Frozen by DVB : February 2000
  - Published by ETSI : August 2000
- MHP 1.1
  - Should be frozen by DVB in June 2001
- Test suite
  - Should be frozen by DVB Q4, 2001

CANAL+ TECHNOLOGIES

When MHP ?

- First MHP compliant terminals are expected Q4, 2001

Who will go first ?

- Likely to start with the retail market (e.g. in Germany)
- In Europe, will be driven by Digital Terrestrial Television
- Official decision to go for MHP : Nordic countries (all networks), Singapore (DTT), Korea (DTS), Australia (DTT)

Which are the brakes ?

- Migration from legacy platforms
- Cost of the terminal
- Interoperability (test suite availability)

CANAL+ TECHNOLOGIES

Conclusion

Technical overview

Schedule

Commercial status

Conclusion

- A good technical solution for iTV (including Internet access)
- Digital TV terminals to go on the retail market
- Solves the content creation issue
- Interactivity as a source of revenues (t-commerce)

- Not yet a commercial success
- Interoperability is the key issue
- 2002 : the year of MHP

OpenCable Applications Platform

Donald P. Dulchinos

Cable Television Laboratories Inc.
D.Dulchinos@cablelabs.com

This presentation describes the OpenCable Applications Platform, a cable industry initiative to develop a common software platform to support a range of interactive television applications and services. OCAP is designed to do so in a way such that these services and applications may run on any cable system in North America, and run on any combination of settop box, television receiver or other device running any operating system software.

The presentation describes the OpenCable hardware specification upon which OCAP is designed to run. Then it provides an architectural overview of the OCAP spec, including its incorporation of presentation engine and execution engine elements. The role of JavaTV APls is described, and the elements of a CableLabs license with Sun for those APls is discussed. Additional focus is given to elements unique to cable industry and customer needs.

The presentation concludes with a comparison of OCAP with DASE and other related specifications, and suggests a roadmap by which these specs can be harmonized.

OpenCable Applications Platform

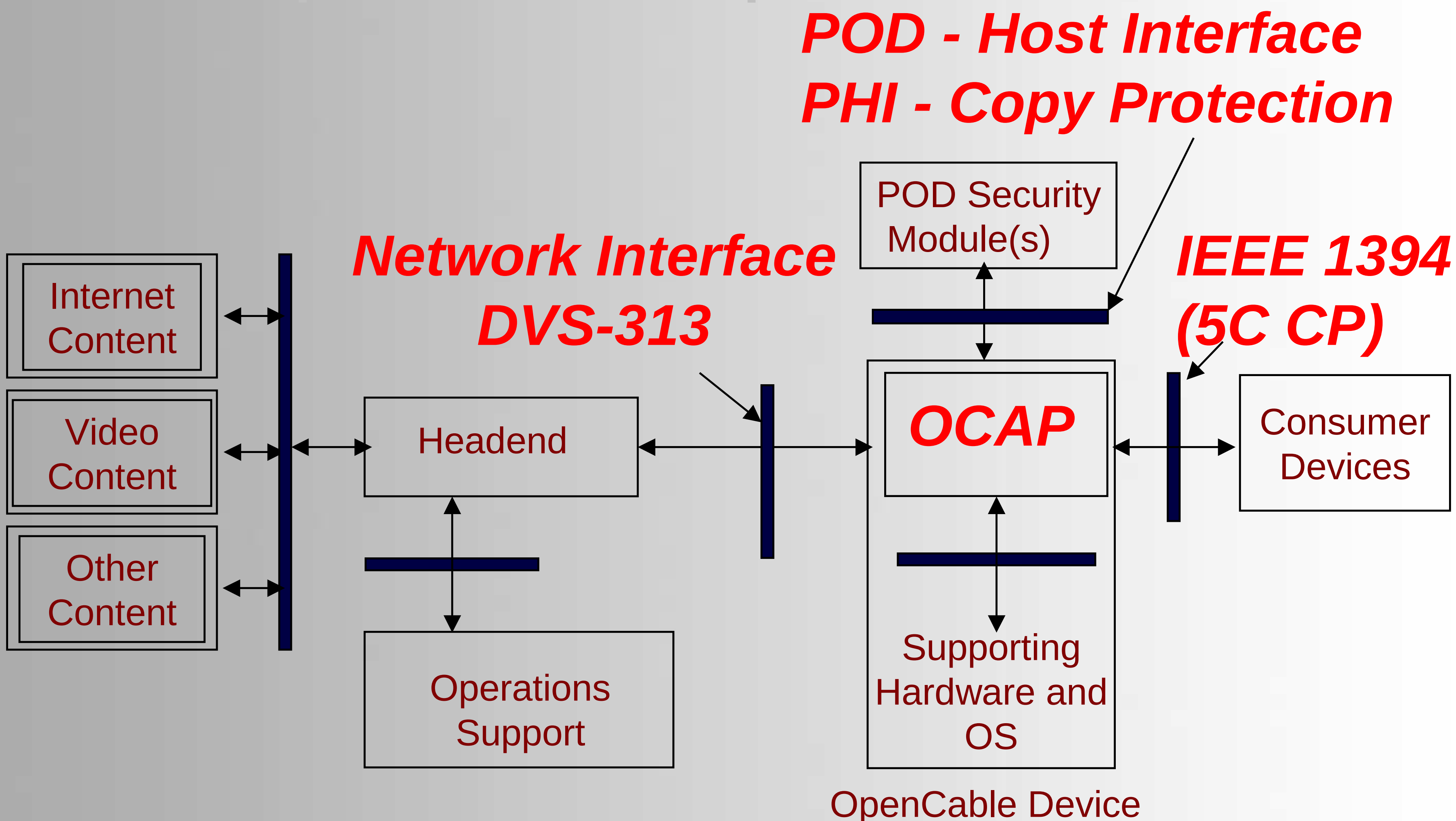
Don Dulchinos
VP, Advanced Platforms and Services
Cable Television Laboratories, Inc

OpenCable Summary

- ***Objectives***
- **Specify the next-generation digital consumer device.**
- **Encourage supplier competition.**
- **Create retail hardware platform.**

- ***Results***
- **Technical specs complete, openly published.**
- **New vendors have entered the industry.**
- **Point-of-deployment security modules available and supported.**

OpenCable Specification



CableLabs

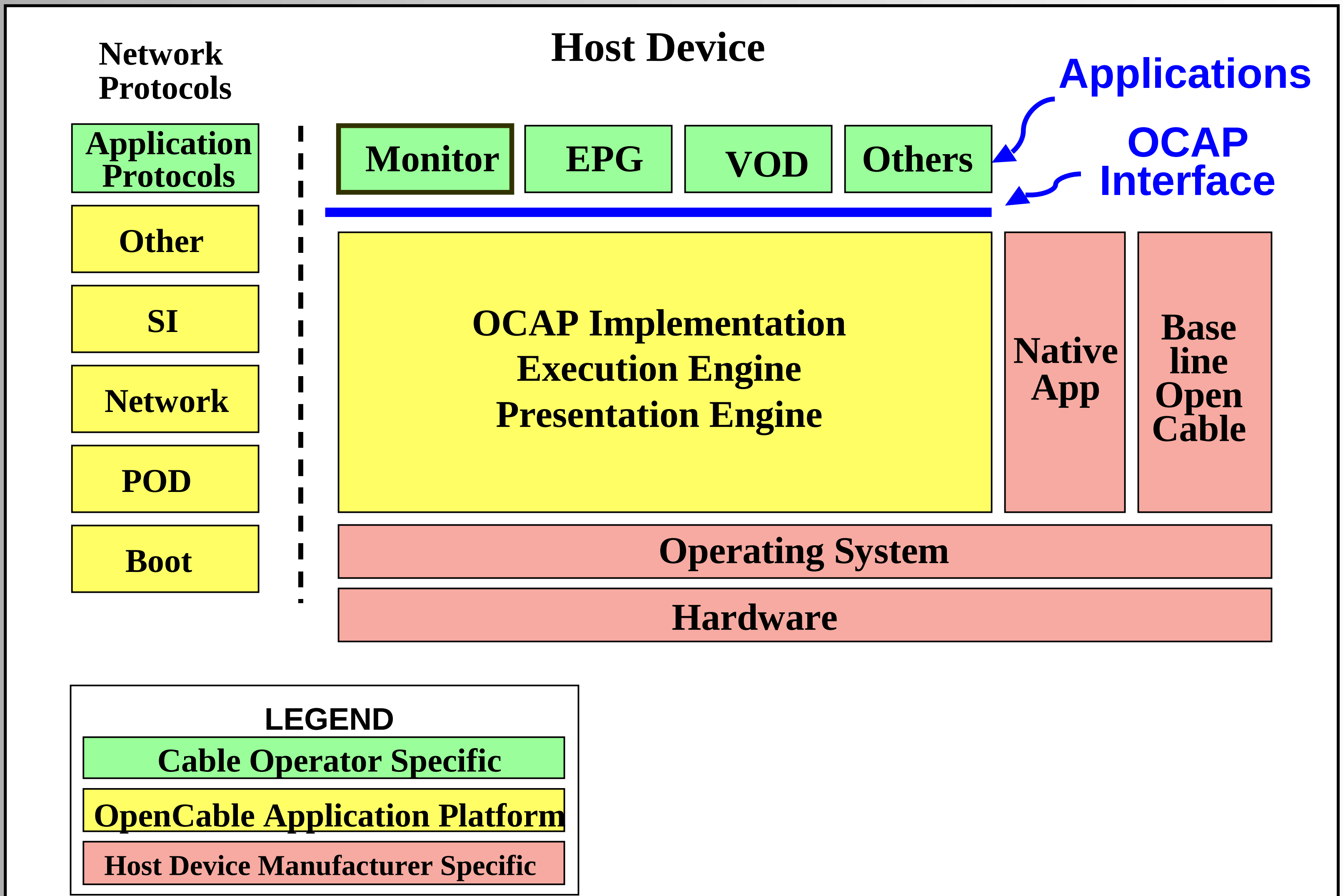
OpenCable Objectives

- **Support retail product developers.**
  - Extend OpenCable to integrated receivers, PVRs, etc.
- **Support interactive service development through common software application platform.**
  - portability of applications, including EPG, IPPV, etc.

OpenCable Application Platform

- **Middleware approach directed by CableLabs Board of Directors.**
  - hardware- and OS-agnostic
- **Business objectives.**
  - enable *service/application* portability
  - preserve supplier diversity
  - encourage innovation.

OpenCable Software Architecture



CableLabs®

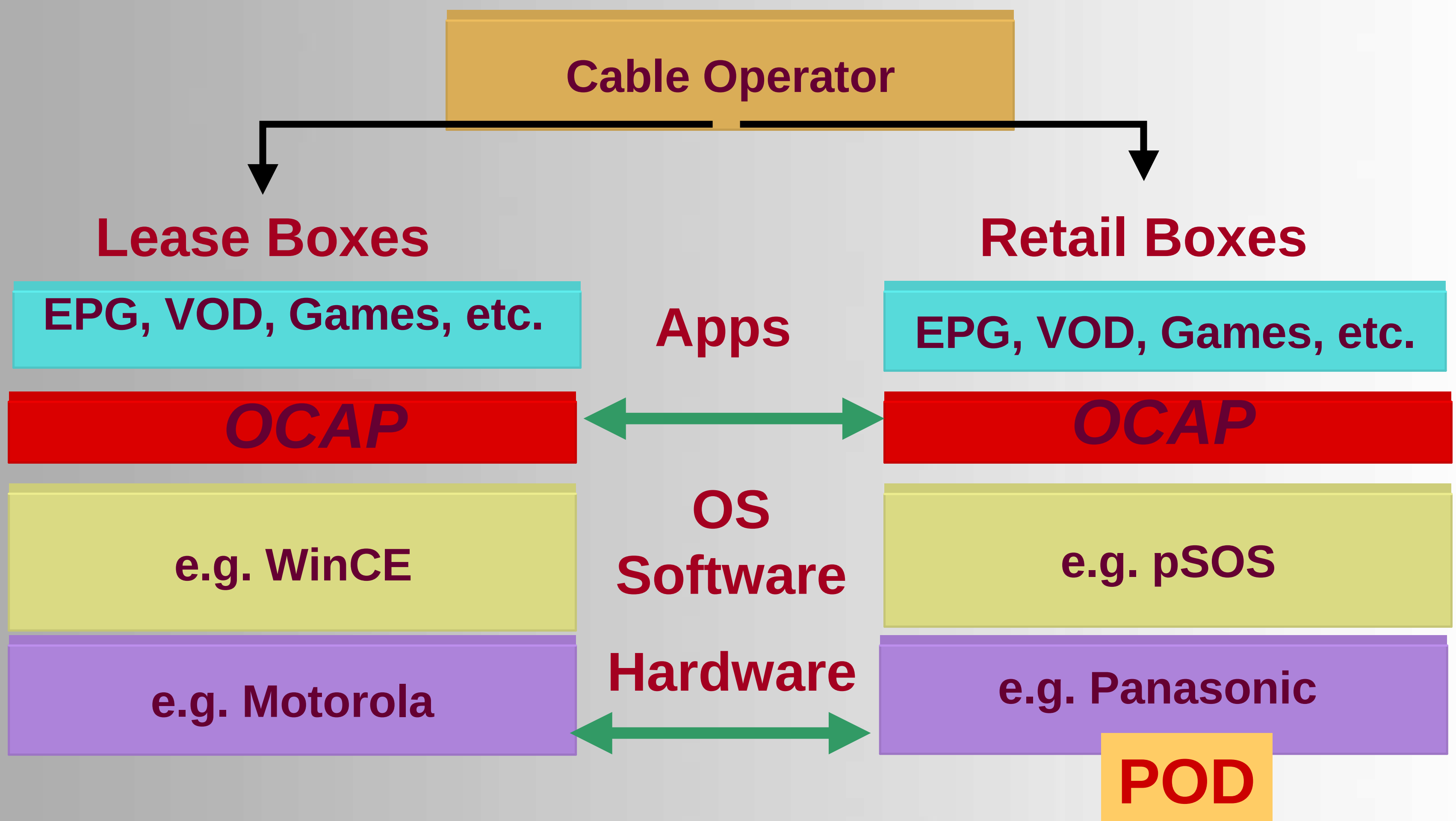
Service Portability

AT&T BROADBAND

TIME WARNER Cable COMMUNICATIONS

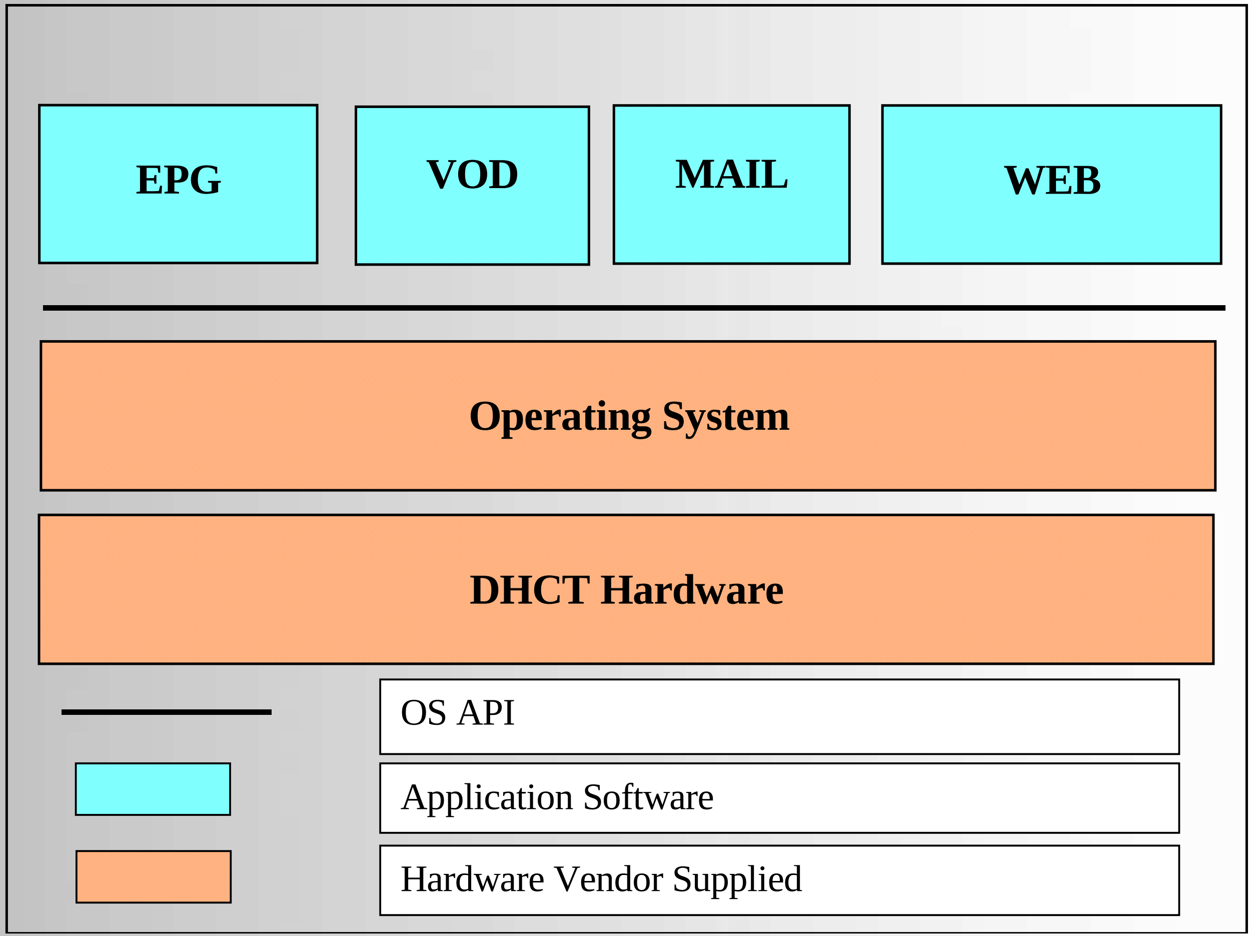
AT&T Broadband		Time Warner Cable
TV Guide, Food.com	Apps	TWC EPG, Food.com
OCAP	↔	*OCAP*
Microsoft	OS Software	PowerTV
Motorola, Philips	Hardware ↔ POD	Scientific Atlanta, Pace

Service Portability

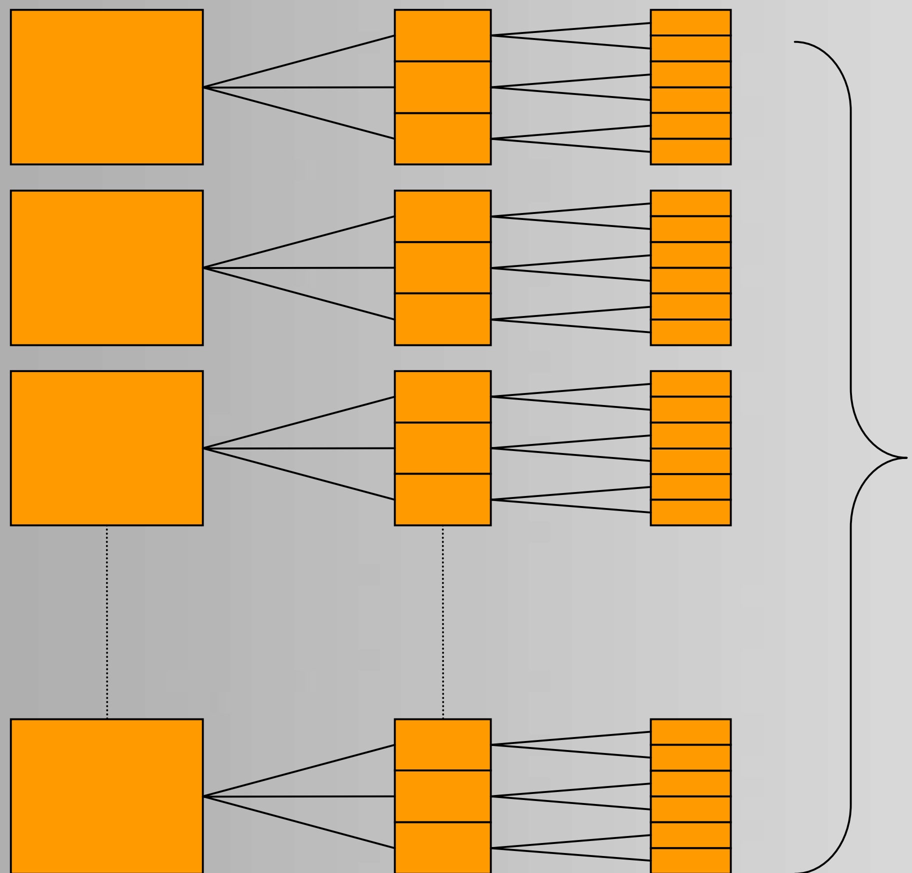


Legacy Software Overview

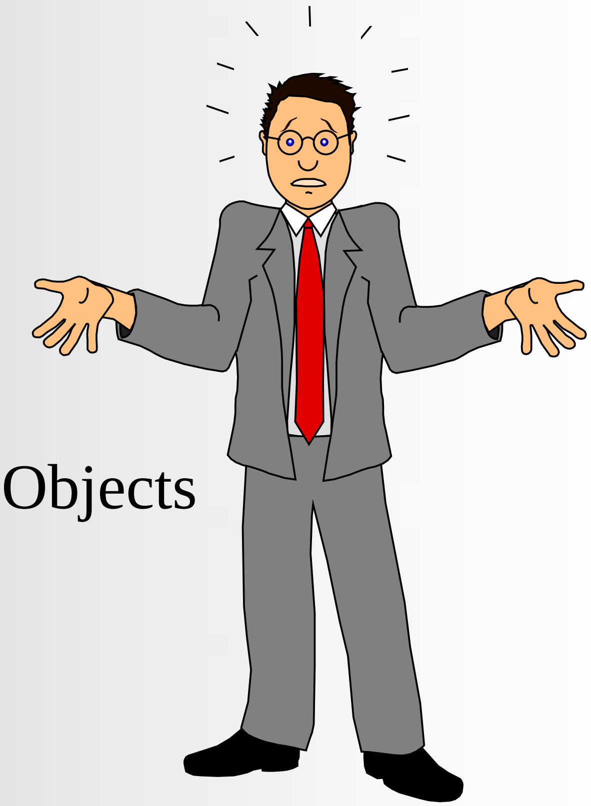
- non portable
- each application must be separately written to the operating system of each type of DHCT **AND** each network



Without Middleware



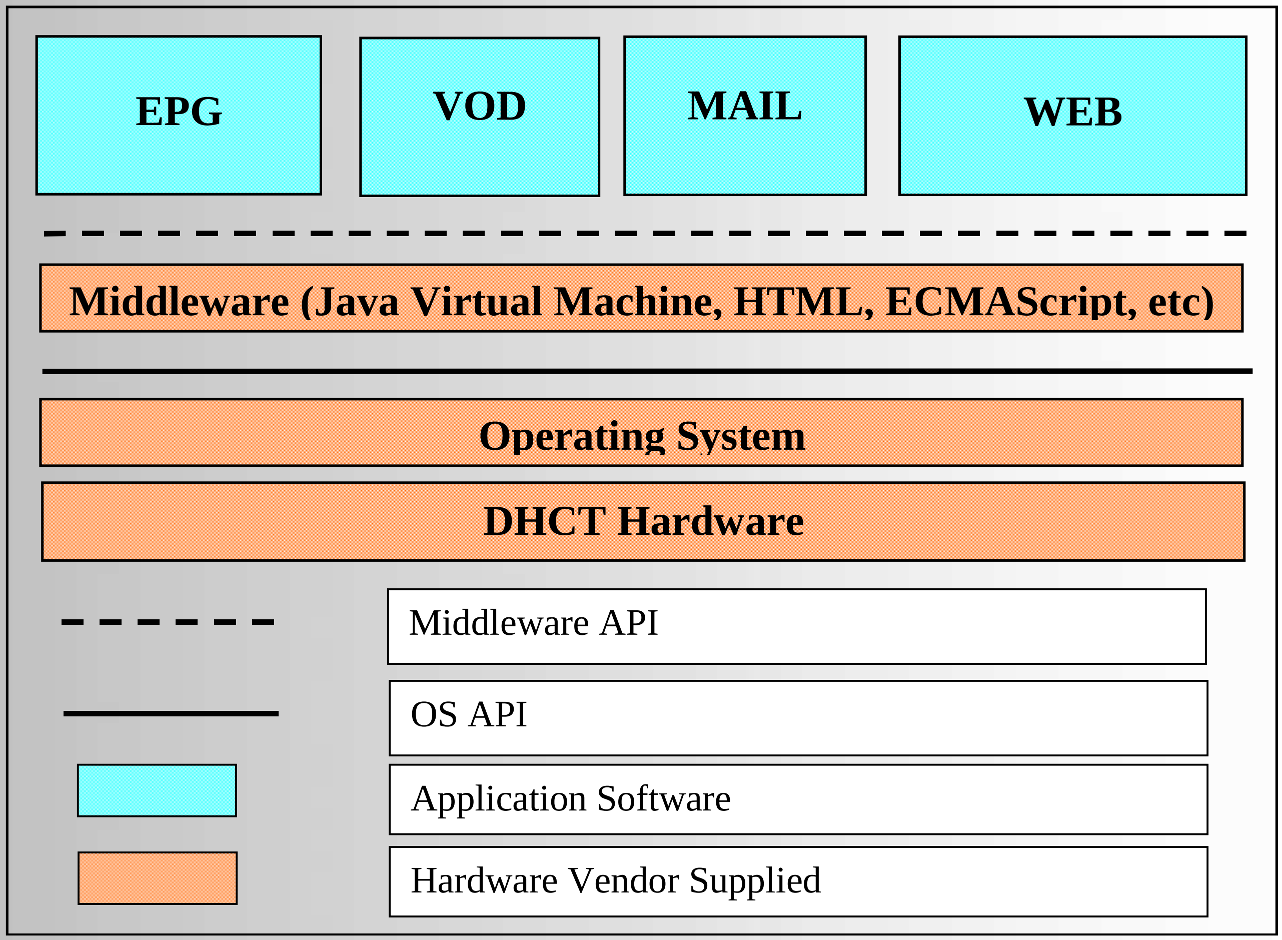
60 SW Objects



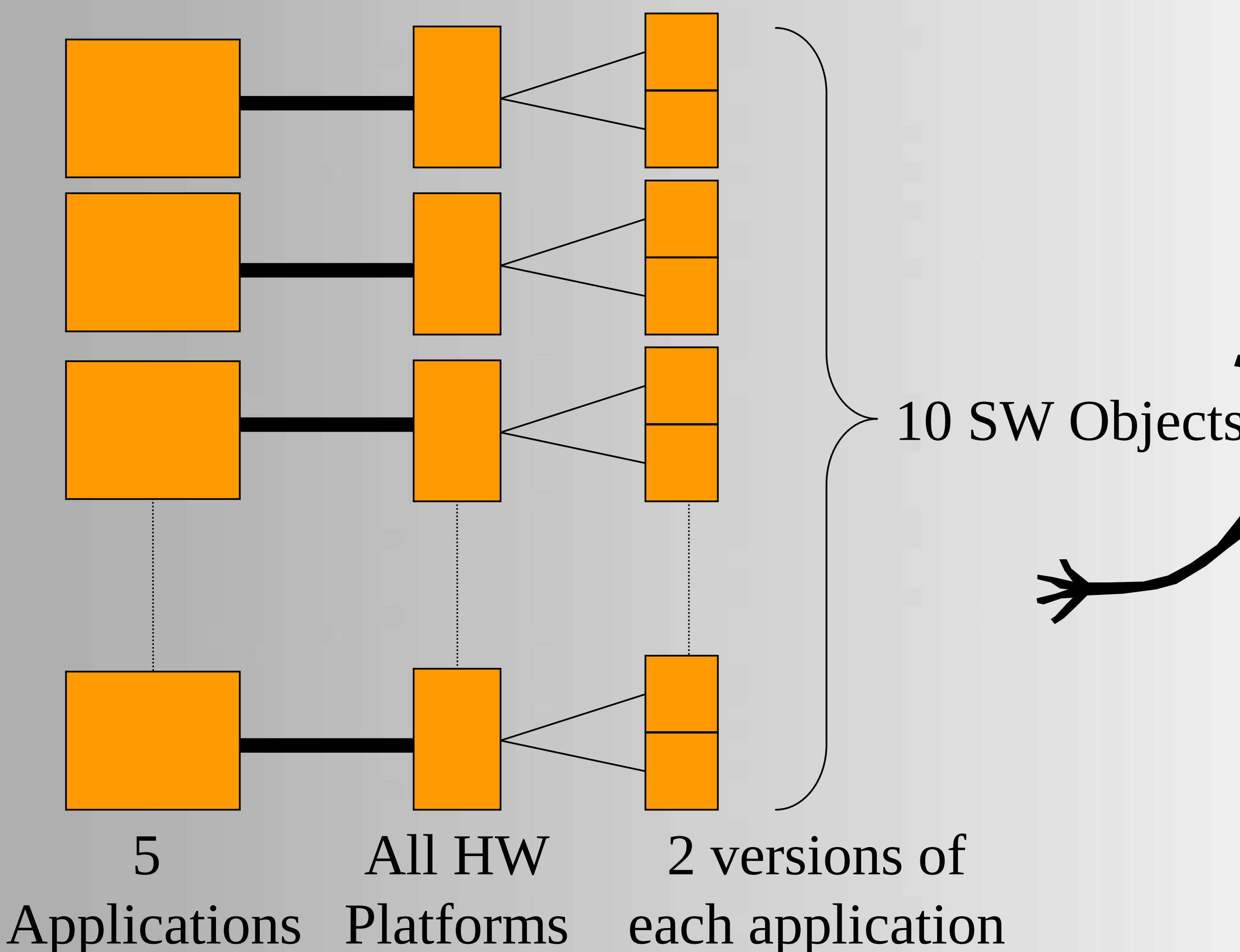
5 Applications | 6 HW Platforms | 2 versions of each application

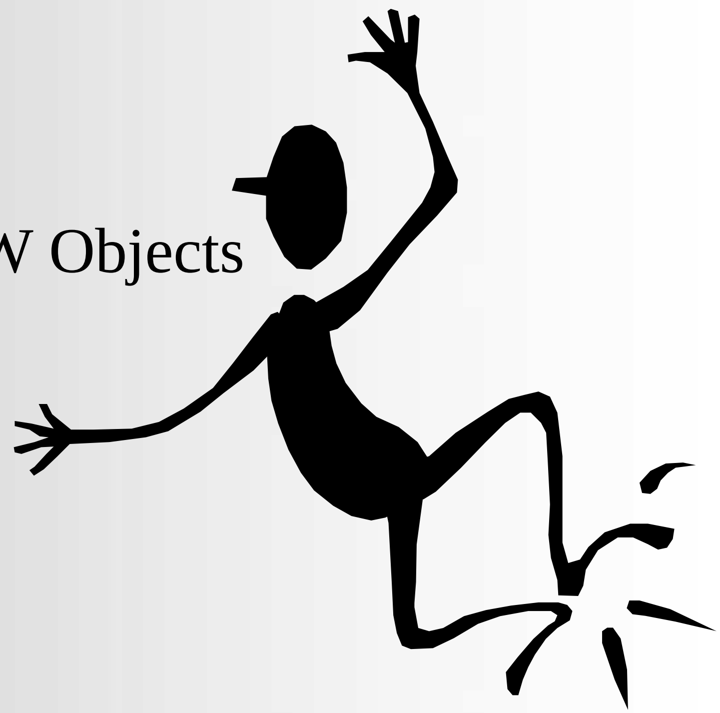
Role of Middleware

- Abstraction layer that makes every platform look the same to the application
- operating system and hardware agnostic



Management -- retail with M/W

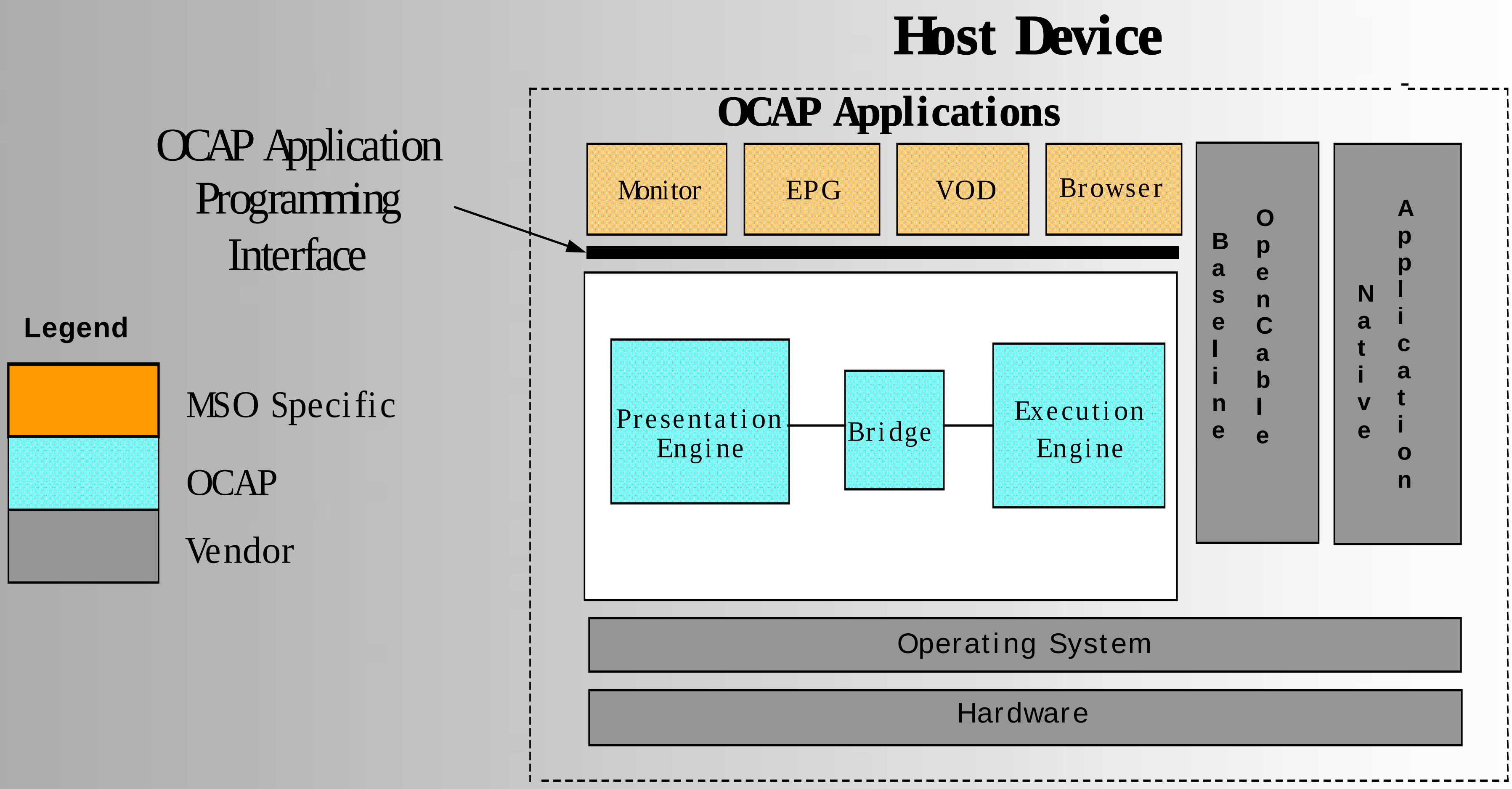




Example Applications

- Electronic Program Guide (EPG)
- Impulse Pay Per View (IPPV)
- Video On Demand (VOD)
- Interactive sports, game shows
- E-mail, Chat, Instant messaging
- Games
- Web Browser: Shopping, Home banking
- Personal Video Recorder (PVR)

OpenCable Applications Platform



OCAP Development History

- RFP process initiated in September 1999
- Proposals returned October 15, 1999
- Review of proposals completed in December 1999
- vendor authors selected
  - Liberate
  - OpenTV
  - Microsoft
  - PowerTV
  - CanalPlus
  - Sun
  - CableLabs, MSOs and Excite@Home
- Specification development began Summer 2000
- Work expedited through the utilization of existing standards and architectures; started with DVB-MHP 1.0

Presentation Engine

- High degree of compliance with DVB-MHP 1.1
- Enable use of tools for developing internet content
- Renders declarative content such as graphics, text, animations and audio
- Consists of
  - HTML 4.01
  - XHTML 1.0
  - CSS 1 and 2
  - ECMAScript 3
  - XML
  - ATVEF

Execution Engine

- Approximately 90% compliant with DVB-MHP 1.0.1
- Java Virtual Machine
- Provides a general application programming environment for networking, file I/O, graphics, etc.
- Security built into the Java architecture
- Provides for full TV application environment (with MHP)
- Features
  - Application management through pJava APIs and XLET controls
  - Service Information and Selection through JavaTV APIs
  - Media control through Java Media Framework
  - Broadcast data through MHP DSMCC APIs
  - Network management and IP data access
  - Extensions from OCAP, HAVi, DAVIC, and DASE

Sun License to CableLabs

- Includes pertinent portions of JavaTV API and related IPR.
- JVM Implementation certified and licensed by CableLabs with no obligation to Sun.
- Sun Technology Compatibility Kit incorporated into OpenCable compliance test suite.
- OCAP can specify the Sun Java Virtual Machine and JavaTV as fundamental components of EE.

Bridge

- Enables browser to take full advantage of resources in STB through the Java APIs.
- Minimizes the use of plug-ins (native applications)
- Permits access by ECMAScript application the Java Class Libraries and Java programs
- Permits access by Java programs to the DOM files

Security

- Application authentication
  - Digital Signatures
  - Certificates
- Permission levels for applications determines access to system resources and APIs--unsigned applications would have lowest permissions
- Encryption to protect private data

Monitor Application

- Optional
- Privileged unbound application
- Cable system-specific
- Control of application life-cycle, resource management, copy protection, reboot, etc.
- Upgradable

OCAP Summary

- Designed for two-way, cable environment.
- Support for wide range of applications and content.
- Portability and uniformity of content display.
- Security and robustness.
- Resource management.
- Open standards.
- Support for developers.

OCAP Status

- Draft specification first release for NDA vendor review - January, 2001.
- Public release OCAP 1.0 - ~June 2001
- Test plan, test environment under development.
- First interoperability testing of applications on different implementations - Sept. 2001.

Harmonization of Spec

- OCAP
- ATSC DASE
- ATVEF
- DVB MHP

- ITU ?

Forum on Cable Interactive Services

CableLabs®

- **Promote cable platform to interactive service/application developers.**
- **Solicit input into OCAP specification from developer viewpoint.**
- **Solicit developer input into interoperability test plans and certification of OCAP implementations.**
- **Recruit service developer contributions in areas of test tools, developer tool kits, training, etc.**

CableLabs

CableLabs®

OpenCable™

Don Dulchinos

VP, Advanced Platforms and Services

d.dulchinos@cablelabs.com

303-661-3803

Audience Measurement Services in a DASE Environment

William Feininger

Nielsen Media Research
William_Feininger@tvratings.com

Nielsen Media Research is the leading provider of television audience measurement and related services in the United States and Canada. Its National People Meter Service provides audience estimates for all national program sources, including broadcast networks, cable networks, Spanish language television, and national syndicators. Local rating services estimate audiences for each of 210 television markets in the U.S., including electronic metered service in 51 local markets. For over 50 years, Nielsen Media Research has provided reports that define the currency for advertising spending on television, which is the basis for free, over-the-air, cable and satellite broadcasting. During these 50 years, much has changed within the television environment, and Nielsen Media Research has adapted accordingly. As the industry enters a new era of in the distribution of entertainment programming via digital television, many new products and services including enhanced/interactive broadcasts, interactive program guides, time-shifted viewing through personal video recorders and T -commerce will be offered to consumers. This presentation will discuss the audience measurement services required for the future. As the Digital Application Software Environment deploys Nielsen Media Research will, once again, be called upon to provide audience measurement services.

Audience Measurement in the DASE Environment

Bill Feininger

Director, Software Metering Technologies

Nielsen Media Research

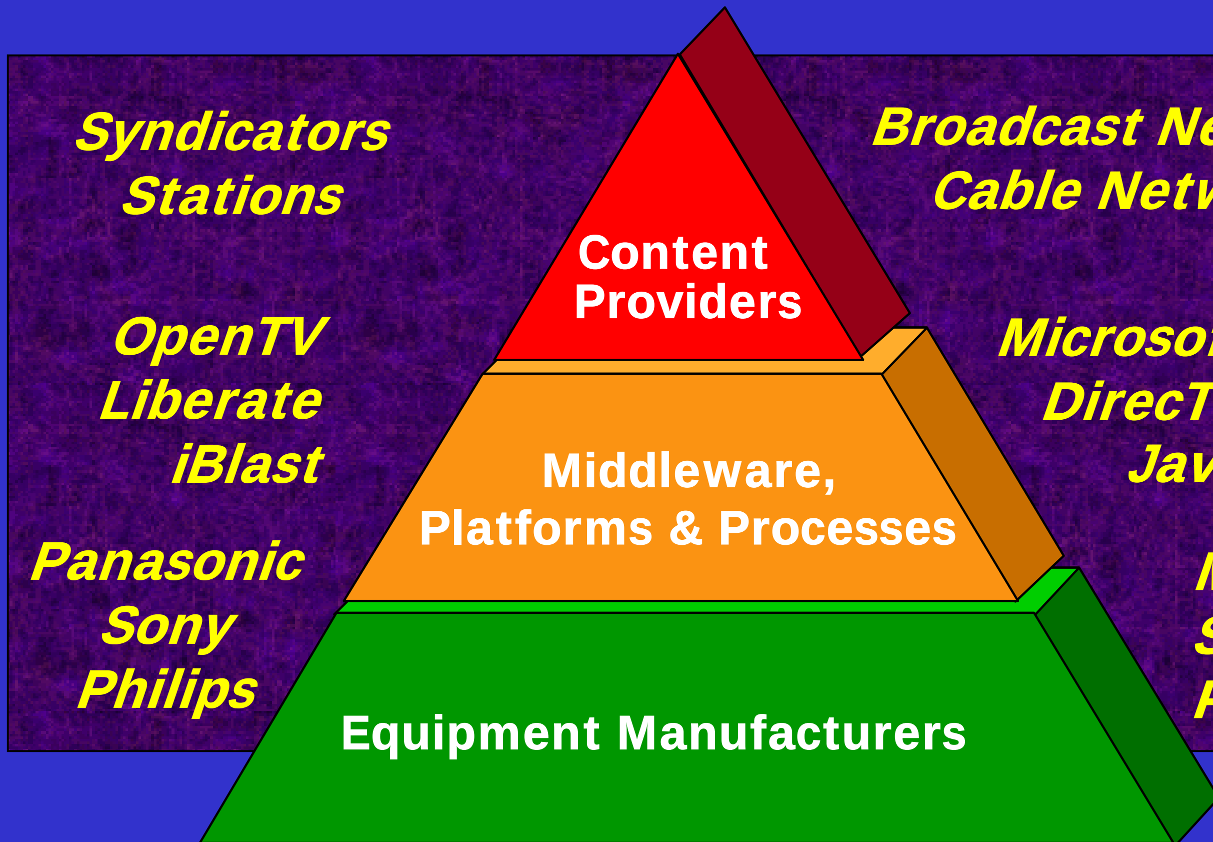
NIELSEN MEDIA RESEARCH

Agenda

- Technologies
- Challenges
- Relationships
- Solutions

A New Paradigm For Television

Many New Players



Desktop Video and Windows Acceleration!
DIGITAL VIDEO
TiVo
replay
Primetime. Anytime.
NDS

Nielsen Hardware/Software Portfolio

Solutions For A New Environment

Analog TV

DBS

Advanced Set Tops

PCTV

Hardware Solutions

Software Engines

Digital Set Tops

PVR's

DTV

Internet

Game Consoles

Interactivity

Analog Television

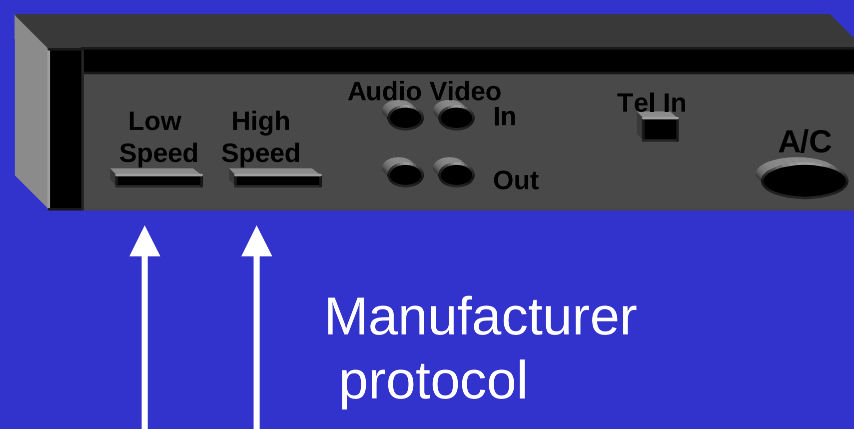
- Reliable frequency-based measurement
  - One 6 MHz frequency = one channel model
  - Frequency probe the local oscillator for channel tuned
  - Automated Measurement Of Lineups (AMOL)
  - “Toolbox” approach with constant improvements

Peoplemeter

MKII

Direct Broadcast Satellite

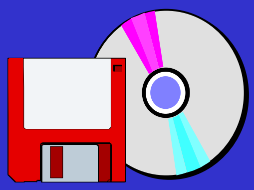
- First entrant to the digital world and to software metering:
  - DirectTV
  - Echostar/Dish
  - Successfully adapted current metering system to measure
  - New equipment/design and new challenges



Manufacturer protocol

PC/TV

- Analog and Digital PC/TV Tuner cards are readily available and inexpensive
- PC/TV tuning allows for variable window size, multiple windows, and multi-task capability
- Next convergent step - TV's with operating systems
- Software solutions are required for this environment
- Field trial underway



Digital Cable

- Analog/Digital Cable Set Boxes
  - Additional digital premium tiers, PPV, sports tiers, etc
  - Motorola and Scientific Atlanta
  - NMR utilizing display readers
  - Working toward a software solution

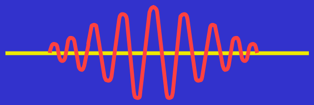
A/P Measurement Techniques

Applying codes to television content at distribution

- AMOL2 Video Code – Source, Date , Time
    - Multi-level ... sec x sec
    - Analog VBI-based
- Audio Code – Source, Date , Time
    - Analog or Digital
    - Suitable for DTV

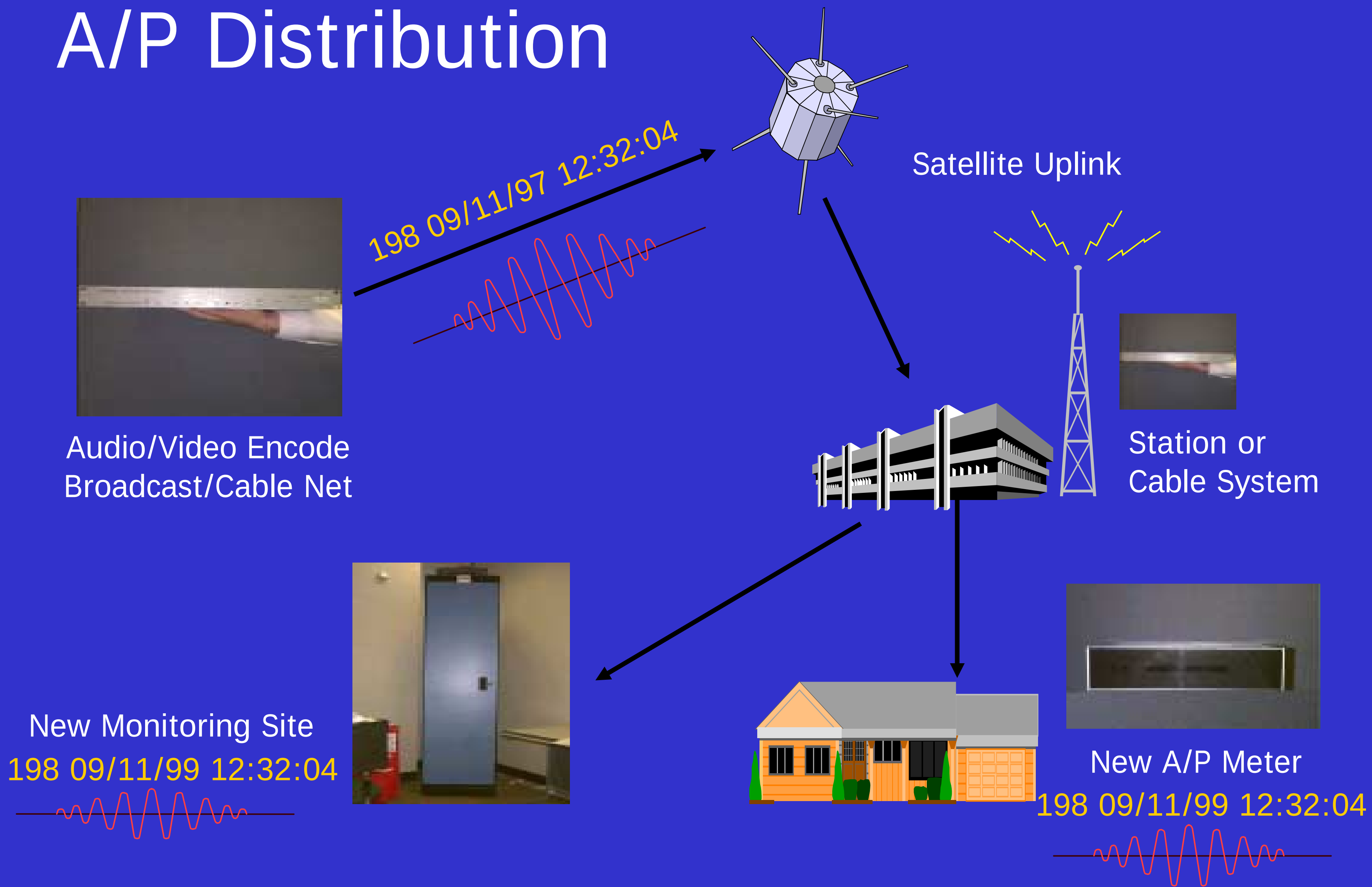
Utilizing Fingerprint information for television content

- Audio Signature – zero-crossing algorithm



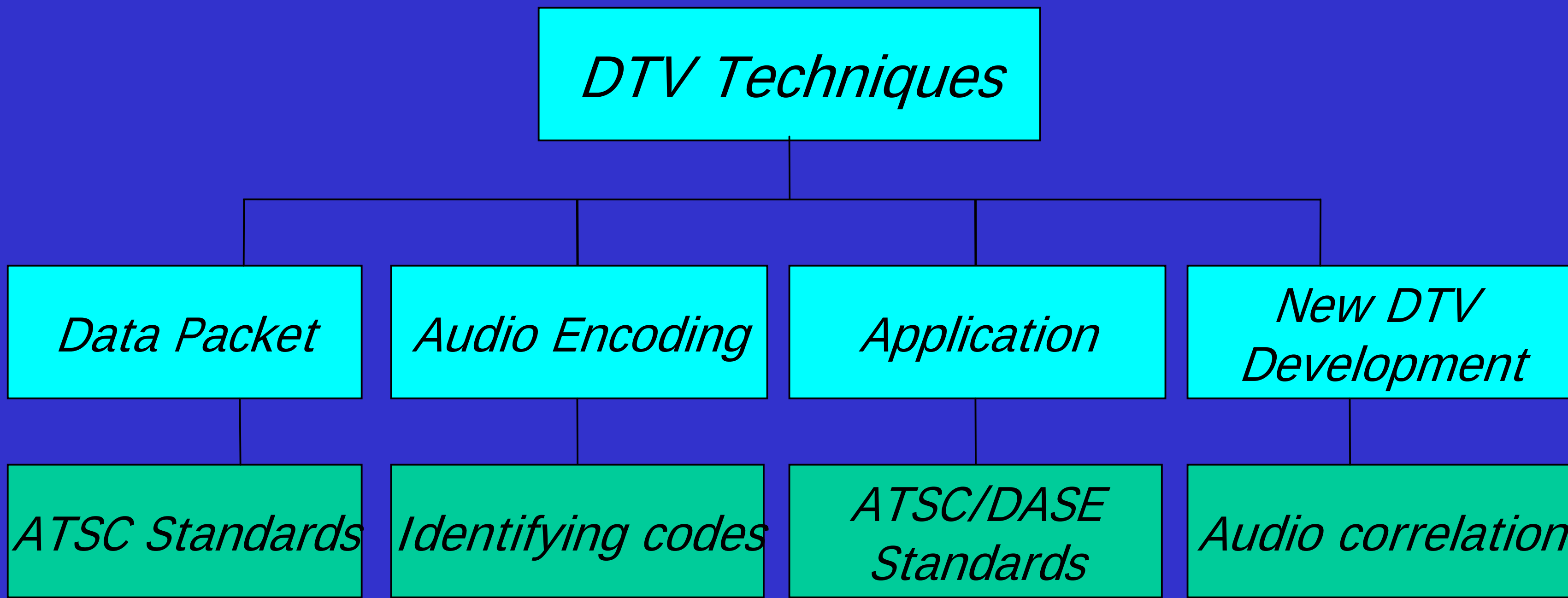
Software measurement solutions

A/P Distribution



Transitional Era – Requires relationships

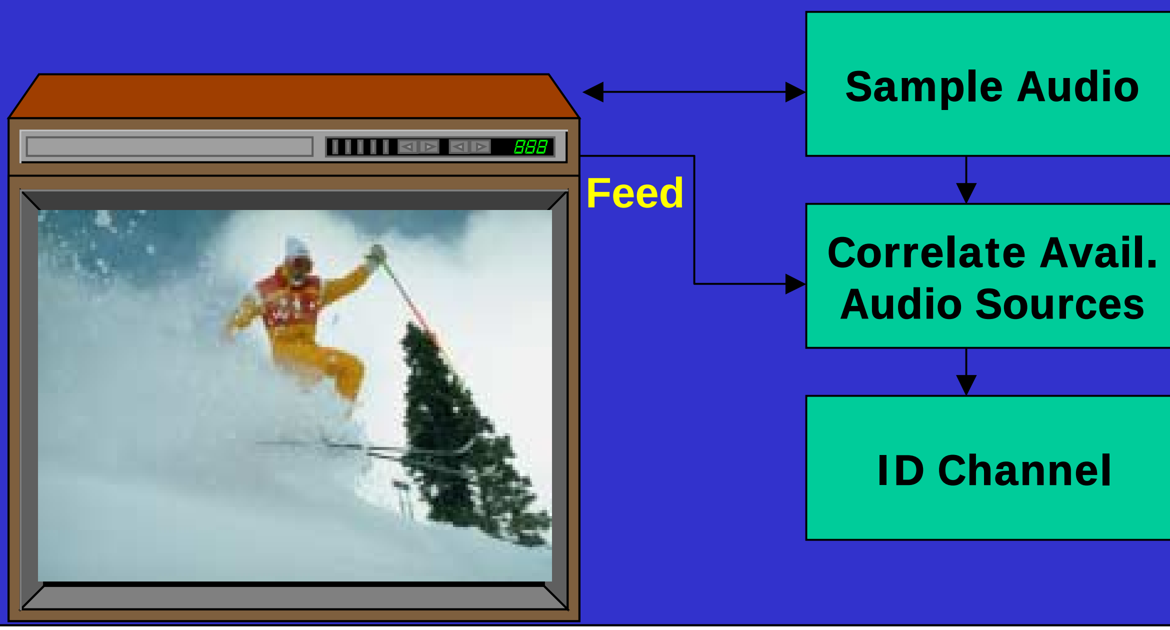
DTV Measurement

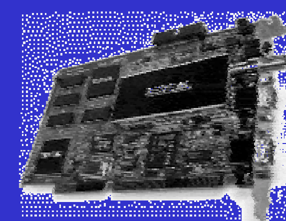


- Frequency measurement and AMOL must change
- New measurement techniques necessary
- New partnerships are required

DTV Development

- NMR-built device for current MKII meter and for our new A/P metering platform
  - Audio correlation, PSIP reader
  - Data packet reader
- Building this approach as a prototype



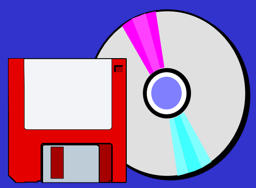


ATSC/DASE

- Powerful set top boxes becoming available
  - Traditional audience measurement
  - Backhaul profiles to advertising community
- Interactivity offers content, ads, and response
  - Mass impressions enhanced with response data by viewers
  - Enhanced measurement of procedural and declarative content for advertisers will be key
- JavaTV applet resident on set-top
  - Applicable to large and small markets
- Uses Service Information interfaces
  - Lists of programs to build EPG's, credit programs, etc.

Software Metering

- New media types
  - Traditional programming (A/V streams)
  - Ad banners (bitmaps)
  - Other streams
  - Applications (procedural content)
  - HTML content (declarative content)
  - Electronic Program Guides
- New collection methods
- New browser-based reporting system

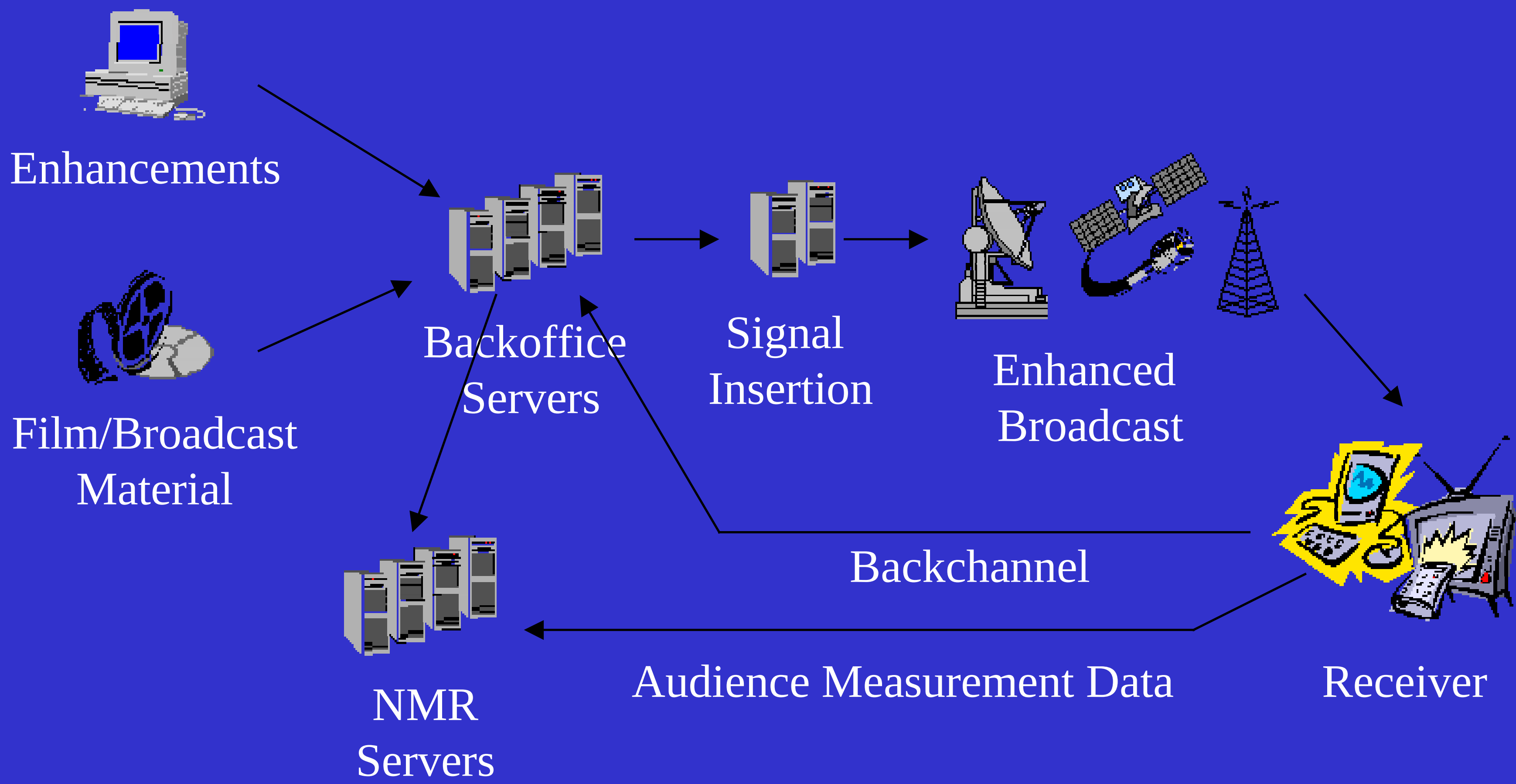


Why Is Such Sophistication Needed?

- Convergence
- Interactivity
- Advanced Analog
- Digital TV
- Personal Video recorders (PVR's)
  - Stand-alone boxes
  - Integrated in STB's and TV's
- Data broadcasting – a way to pay for digital
- PPV and VOD – at last an economic reality
- STB's as networking gateways

Technology deployment is making the environment far more complex …
Thus, a broad array of measurement tools and relationships is needed

New Measurement System



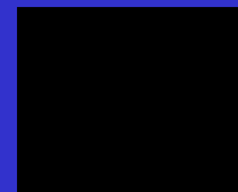
What Lies Ahead

- DTV has arrived and is hitting the early adopters/high income/high education demos first
  - NMR audio correlation for DTV will be key to measurement with MKII and/or A/P
  - Data packet would be efficient
  - Audio encoding is an excellent approach to measurement
  - DASE-based measurement solution

Parting Thoughts

- Planning for the future
- Build business & technology relationships
- Integrating old and new systems
- Software/hardware solutions
- Leverage innovations
- Nielsen is making the necessary investments
- Measurement requirements are changing, and we need industry-wide input
- Partnerships (both technical and business) are key

Hardware Solutions



Software Engines

Questions ?

Managed Media Service Platfonn

Ravi Gauba and Klaus Hofrichter

Sony Electronics
3300 Zanker Road, San Jose CA 95134
TEL: +1-408-955-4351, FAX: +1-408-955-3030, E-mail: rgauba@arch.sel.sony.com

The *Managed Media Service Platform (MMSP)* is being researched and prototyped in the Sony US Research Labs. It is a service platform based on local audio-video hard disks with multiple TV tuners or an always-on broadband Internet connection to support a new breed of media services. The fundamental concept of the *MMSP* is that the client device hosts one or more media cartridges. Each cartridge contains separate disk space and a tuner, and is individually dedicated to one specific service provider who manages the content on the customer's disk. The content is accompanied with descriptive metadata, which enables selection of content on demand, based on the viewer preferences.

The *MMSP* research addresses the design and implementation of a service environment, which takes advantage of digital audio-video technology to offer new kinds of viewer-centric media services. The *MMSP* client device stores the TV broadcast program or an online media stream continuously on an A V hard disk, along with metadata, which describe the content. The Metadata helps the service provider maintain and control the storage of content on the client, based on policies set by the service provider and using the viewers profile. The viewers profile is created and maintained by the profiling engine, which tracks the users viewing habits and his choice of programs. The metadata along with the viewers profile provides fine-grain access to the delivered content and is used during playback to present the content in an interactive, meaningful, television-oriented, personalized form.

Several service concepts have been prototyped in the course of the research: Custom News Service provides news stories that are always up to date and the lineup is presented based on the viewer preferences. The service also provides multiple viewing modes like condensed viewing, detail viewing or normal viewing described in detail in the paper. Movie Rental Service stores movies based on the viewer's preferences. It allows the viewer to unlock these movies and watch them instantaneously. The News scenario and the movie rental are only some of the possible application of a *MMSP* system.

Other applications such as TV games, education, are also feasible.

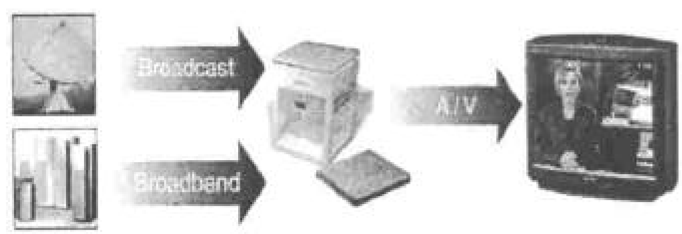


Figure I: The *Managed Media Service Platform*

Figure I above sketches the system configuration: the *MMSP* cartridge hosts the tuner for broadcast content, an always-on broadband modem and a dedicated hard disk in a service cartridge. Cartridges are plugged and installed into the *MMSP* cartridge tower. After the installation the cartridges begin automatic content capture from the assigned service provider.

The full paper about the *MMSP* introduces the technologies and business concepts that are involved in the development of the prototype. It covers in particular the design of the platform, the metadata and describes the show flow engine, which processes the metadata and the viewer profile to generate the presentation. The paper also addresses and compares the richer capabilities and new applications provided by *MMSP* when compared to today's digital video recording devices. *MMSP* enables new applications, business opportunities and better service quality.

A Managed Media Service Platform

Ravi Gauba and Klaus Hofrichter

Sony Electronics
3300 Zanker Road, San Jose, CA 95134
TEL: +1-408-955-4351, FAX: +1-408-955-3030, E-mail:
rgauba@arch.sel.sony.com

Abstract

The *Managed Media Service Platform (MMSP)* is a research project of the Sony US Research Laboratories. It is a software platform designed for residential customers that exploits the use of local audio-video hard disks with multiple TV tuners or an always-on broadband Internet adapter to support a new breed of media services. An *MMSP* client comprises a base station and one or more plug-in media cartridges, each belonging to a specific service provider. A media cartridge contains an audio-video hard disk to store data (media assets, metadata, code) and a TV tuner or a broadband Internet adapter to receive the data. A service provider is associated with a specific cartridge and manages the data stored on the customer's disk. Media assets belonging to a service are presented on a TV display under consideration of the viewer preferences.

1.0 Introduction

The *Managed Media Service Platform* (*MMSP*) addresses the design and implementation of a service environment, which takes advantage of digital audio-video technology to offer new kinds of viewer-centric media services. The work described in this paper is based on lab research taking technologies such as Digital TV, broadband Internet, A/V storage and metadata into account.

MMSP enables a personalized television experience by leveraging non-linear content and editorial structure for rich interactive content presentation in the home. This "fine-grain programming" along with content provider-driven storage management is a key differentiator of the *MMSP* compared to traditional A/V storage (e.g., PVRs). Since the content provider has editorial control and has exclusive access to the allocated storage on the service cartridge, the provider can manage the content, storing, removing, and updating media via either one-way broadcast or Internet delivery.

The goal of the project is to specify an architecture for deployment of audio/video services that use service management driven by metadata and user preferences. The *MMSP* provides an environment for managing A/V services along with their content (A/V data) and data assets (e.g. audio, text, graphics objects, etc) in the customers' home. The system is based on an A/V media cache, which is controlled by service providers. It allows linking the content and assets together to provide a TV-like, video centric viewing experience. The platform addresses content delivery and service management issues and can be used to integrate broadcast and broadband data delivery. The *MMSP* client device stores a specific TV broadcast program or a broadband media stream continuously on an AV hard disk, along with metadata, which describe the content. The software deployed by the service provider on the client device uses metadata to maintain and control the storage of content, based on policies set by the service provider and using the viewer's profile.

The viewer's profile is created and maintained within the *MMSP* system, which tracks the user's viewing behavior and content selection pattern. Metadata associated with the content provides fine-grain information about the delivered content to populate the viewer profile. The viewer profile along with service specific configuration of *MMSP* software components is used during playback to select content for presentation.

MMSP is intended to be implemented on a small, inexpensive base device that supports plug-in cartridges. Each cartridge contains a tuner and appropriate A/V disk storage for specific services. A consumer purchases the services individually, comparable to a game cartridge or a DVD. The costs of the *MMSP* base station and the cartridges may be subsidized, since the service delivery may include a service subscription fee.

The following sections of the paper cover the overall system architecture and discuss the main components of the *MMSP*. Some example services developed for the prototype are discussed. This paper does not cover the content creation and authoring aspects.

2.0 System Overview

Although the prototype implementation of the *MMSP* system is done in software only, it is necessary here to describe the envisioned hardware components as well. The design of the client device hardware plays an important role in the overall system, since the A/V cartridges manifest the resources linked to a certain service. The design described below implies that a dedicated cartridge represents a specific service. Alternatively, a single cartridge with larger capacity may provide partitions to host multiple services, but this approach would limit the flexibility and requires negotiating resource allocation across service providers.

Figure 1: Data distribution with the MMSP

The envisioned *MMSP* base device hosts one or more A/V media cartridges. Figure 1 shows broadcast and broadband content feeds from the service providers to the *MMSP* service cartridges. The data streams consist of a mixture of A/V content, metadata, executable code and media assets. All incoming data is temporarily cached until the local cache management matches it against the local user profile. Irrelevant and obsolete content is removed from the disk. This strategy enables the service provider to use the same content stream for many clients, which is particularly important for broadcast content distribution. The service provider may or may not use individual data streams with pre-selected content for each client in a broadband distribution environment.

The *MMSP* base provides A/V output jacks to connect to a TV, and a set of service specific input connectors such as Ethernet for broadband Internet access, and RF broadcast TV input plugs. An infrared remote control is used to control the service access. The cartridges are connected to the system through a backplane that provides power, and analog and digital signals. The cartridges run independent of each other to capture and maintain the incoming content, but only one cartridge provides an output signal at any given time. The base station provides graphic and video rendering capability for the active cartridge.

3.0 Service Platform

The *MMSP* is designed to provide enhanced A/V services, which take advantage of metadata associated with content and local storage with service provider driven media management. The services are able to adapt dynamically to the customer's viewing behavior. The *MMSP* base station comprises several software components, including a resource manager, a rendering engine, a broadband adapter, and communication facilities for the cartridges.

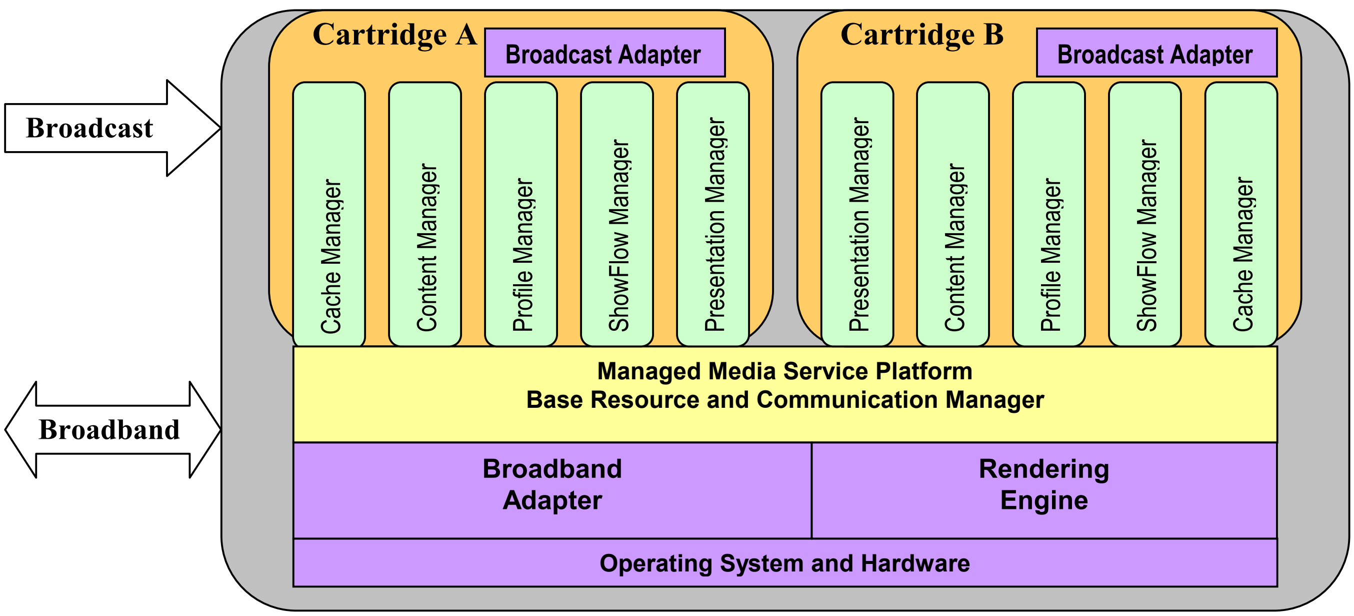


Figure 2: MMSP client architecture

Each cartridge connected to the base station runs its own service specific software for content storage and management, viewer profile, content selection (so called ShowFlow) and presentation. A service provider may decide to implement any one or all of these components in a proprietary manner and ship it with the cartridge or to use default components that are customized by a set of configuration rules. This allows the service provider to differentiate their service functionality or to ease deployment. The shared software components running on the base station offer a defined API, which is used to access the presentation hardware and broadband Internet adapter. The software components in the cartridge and the base communicate with each other using an asynchronous event mechanism.

The metadata provided along with the content by the service provider is used by the *MMSP* components such as the content manager and the profile manager to decide how content is stored and presented. The metadata schema used by the *MMSP* is a subset of MPEG 7 metadata along with custom, service specific extensions. The service provider can also configure and control the client devices using command and configuration scripts, which are delivered to the client device along with the content. These scripts are used by the *MMSP* to configure the behavior of components like the cache manager.

The metadata is delivered to the client along with the content and may use the broadband delivery mechanism or a broadcast channel. The content metadata is created during the content production. Most of the attributes can be imported from existing content databases. The service provider reuses the data structures and adds *MMSP* specific information. The following sections provide some details about the core components that drive the *MMSP* system. Content metadata enable the selection of content that fits the viewer's needs. The content manager uses the metadata to determine the relevance of the available media assets. The viewer profile provides information about the user's preferences. The ShowFlow component controls the presentation engine to visualize the content in a personalized way.

3.1 Cache and Content Manager

The cache manager is responsible for management of storage of A/V content, metadata and other assets delivered by the service provider. It evaluates incoming data using the associated metadata to determine if it fits the viewer's profile. If required, expired or less relevant content is removed from storage to make room for new assets. Usually, storage will be kept full with content, and each storage operation requires deleting other content. The algorithm to identify the obsolete content is controlled by cache manager policies, which are specified by the service provider. Since the complete storage capacity of a cartridge is reserved for a specific service, the service provider is able to change and update the caching policies. This approach does not require any user intervention. Moreover, most of the use cases based on the *MMSP* system do not allow for direct access of the content by the viewer. Instead, the service is comparable with a newspaper, where new content is delivered regularly, without direct influence of the subscriber over the delivered content. It is up to the service provider to exploit the user profile as much as needed. For example, the provider of a TV News application may on the one hand recognize viewer preferences (e.g. local politics, sports), but at the same time it is possible to maintain editorial guidance by enforcing the presentation of the top-story of the day, even when the profile setting does not cover this. The cache manager is also able to update specific content. For example, the service provider may instruct the client device to overwrite an obsolete piece of content if an updated version is available. This is a typical procedure in a TV News application, where new information is replacing outdated data, for example a weather forecast video clip.

The service provider controls the content on the client device using the content manager. The service provider sends instructions regarding the content handling in the content metadata. The content manager then interprets the rules. The content manager validates, tracks, and authorizes content stored in the cache. It maintains a list of content and assets that are available for viewing. The content manager keeps track of all the content and checks if it is valid for viewing based on time and authorization. For example, after a movie is rented from a movie rental service, the content manager maintains a record of how much of the movie is viewed and how may times it is possible to watch the movie.

3.2 Profile Manager

The profile manager works with both the explicit viewer choices and the implicit viewer behavior. In the *MMSP* we define the explicit configurations done by the viewer and the explicit likes and dislikes defined by the viewer as viewer preferences. While the profile is the implicit viewer behavior learnt by tracking the viewer usage pattern. The profile manager tracks the viewer's content usage and generates the profile using the metadata associated with the content. The service provider defines the impact (or weight) that viewing of a particular content generates. When a viewer indicates interest in a program by watching it, the system changes the user profile on one or multiple content categories based on the weight assigned to the each category of the active content.

The profile manager uses statistics and analysis of viewing patterns and explicit indication from viewer to learn how much a certain content type is desired by a viewer. For example, a viewer can define in a movie rental service a preference for comedy movies, but never watch available comedy movies. This would cause implicit lowering of the weight of the particular category, without removing the generic "comedy" preference.

3.3 ShowFlow and Presentation Manager

The ShowFlow manager selects the content that is used for presentation to the viewer. It generates a list of content that is passed to the presentation manager. The list is determined after evaluation of the viewer preferences, the viewer profile and the metadata of the available content. The evaluation process is either controlled by a set of rules or custom software delivered by the service provider.

The content provider may supply specific metadata tags for the content to support different viewing modes. If these tags are available, the viewer may select one of these viewing modes. Four different modes are available: normal, condensed, highlight and interactive. Each mode provides a different view to the content with varying level of detail and interactivity. The normal mode is the default mode for all content to be presented. The condensed viewing mode is used for an abridged version of the presentation. In condensed viewing mode the viewer will get an overview about the content but sections with less importance are not shown. The content is even more reduced in the highlight mode, where the viewer only sees parts of the presentation that stand out, with the consequence that the context of the presented pieces might get lost. The interactive mode gives the viewer the option to access more details and more complex navigation.

For example in a movie download application, the normal mode provides an interactive menu to select and watch a movie. When selecting the condensed mode the movie presentation is shorter comparable to an abstract, but the full storyline is still present. In the highlight mode only the most exciting scenes will be shown, with little association between them. The interactive mode presentation allows activating options such as the directors' comments on a particular scene, or additional information on the background music etc.

The ShowFlow manager interacts closely with the presentation manager, which uses the rendering engine to display the content to the viewer. The presentation manager generates instructions for the rendering engine that define how the content is presented and how the viewer can interact with the application. The service provider has full control over the presentation since the ShowFlow and presentation managers are service-specific components. The rendering engine can present HTML or run Java bytecode.

4.0 Example

A number of different services have been implemented to demonstrate the *MMSP* architecture, including Customized TV News and Movie Rental. These services use different kinds of business models, i.e. subscription and pay-per-view.

4.1 Customized TV News Service

Customized TV News is well suited for the *MMSP*. This service is intended to be based on a subscription business model where the viewer pays a monthly fee for enhanced, personalized, interactive TV news content. The content is offered to the viewer according to the local preferences in a TV-oriented way: The presentation is stream oriented and does not require the viewer to become active as the content is presented. Instead, the presentation manager arranges the separate news clips automatically to meet the viewing pattern of the viewer.

The viewer sees the headlines in the beginning of the presentation in form of an animated table of contents. The table of content consists of a number of small teasers for the upcoming stories. The order of the news clips depends on a number of factors, including the viewer's preferences, the viewing history, and the content provider's editorial specifications. The latter factor enables the content provider to indicate important stories, which might enforce presentation of stories even if the preferences would not cover this particular story. The presentation of the table of content can be used by the viewer to express interest or disinterest in the separate stories. This may result in a change in the actual presentation order and the preference settings of the viewer.

The actual news stories are shown after the table of content. The ShowFlow manager arranges the stories according the viewer's rating, the available time and the preferences. The stories are presented in a sequence, with optional additional information such as related material (such as subtitles) or custom data such as stock-tickers.

The viewer can navigate in the story lineup, but this is not required as the presentation automatically moves forward from clip to clip. This resembles the format of a moderated TV news show, with the added value that the selection of the content is highly personalized.

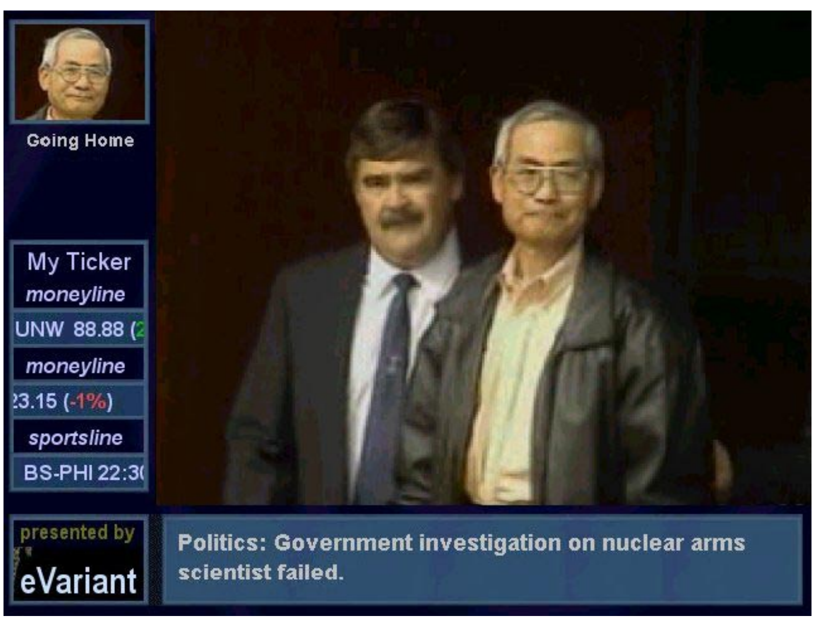


Figure 3: Interactive TV News with metadata presentation

4.2 Movie Rental Service

The Movie Rental Service is designed to support a pay-per-view or rental business model. The client cartridge for this service stores a selection of movies fitting the viewer preferences. The service provider pushes movie files to the cartridge. The viewer may then rent one or more of these movies, which is immediately available since it was downloaded in advance. This service requires a back channel for the payment transaction.

The content manager system keeps track of the rented movies and removes them once the renting period is over. The viewer can also mark movies that are available without actually renting them to prevent automatic removal. This allows the movies to remain in storage for a longer time.

The application supports multiple users with different profiles. This enables the implementation of features like parental control and similar viewer-specific functions that restrict the types of movies available to a viewer. Figure 4 shows how these movies are presented along with trailers, images and additional information on the movie.

Figure 4: Movie Rental Service

5.0 Conclusion

The *MMSP* is a software platform designed for a set top device. It provides support for personalizable interactive applications. The *MMSP* software is envisioned to be used in a base device that supports cartridges with a large A/V disk, tuner and service specific software. The platform provides support for the cartridges and for audio/video service management and control. The *MMSP* enables through service specific reservation of A/V storage capacity new types of applications, branding and business opportunities. Service providers manage a device in the customer's home, as well as the content and the means of presentation of the content. This creates a close relationship between a specific provider and the customer. The service providers manage content on the customer's cartridge based on viewer preferences, content metadata and usage history.

6.0 References

[1] TV-Anytime Forum: http://www.tv-anytime.org/
[2] MPEG 7: http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm
[3] DASE API: http://www.dase.nist.gov/documentation.html
[4] HAVi UI: http://www.havi.org/
[5] MHP: http://www.mhp.org/

SMPTE 292M VANC
A Practical Way to Handle 'Sticky' Data

Jim Carruthers

Norpak Corporation
10 Hearst Way
Kanata, ON, Canada, K2L 2P4
jim@norpak.ca

The focus of much of the DASE and ATSC data broadcast standards is, as it should be, on emission. However, for DASE to succeed we need to also consider the production chain prior to the point of emission. While content can be inserted at the point of emission, much of the most compelling interactive content will be generated as an integral part of the creative process. The technical solution needs to take into account how content is created, timed to video/audio, stored, and retrieved in the end-to-end chain, including those locales where the video is in its uncompressed baseband form. The V ANC portion of the SMPTE 292M signal offers a simple yet powerful way to deliver data while using proven broadcast procedures and processes.

Since the VANC is part of the 292 signal the combined video, audio and data signal can be switched, routed, stored and recalled using familiar processes. Captioning, metadata and interactive content injection continue to be done at the same point and in the same way in the business process. The 292 V ANC space has significant capacity.

While the VANC can carry any type of content it is particularly useful for 'sticky' content such as DASE data, which:

- Originates as part of the original creative process and is integral to the overall viewer experience.
- Needs to 'stick' with the video and sound as the content makes its way from the creator, through post-production, to the network headend and is distributed to the point of broadcast.
- Must be transparently stored and retrieved without the complications of locating it on a server somewhere. Program delays of seconds, hours and even days are commonplace. In an environment where content rarely makes money on first run, continued playout of the complete experience, often at short notice, is a business imperative.

SMPTE 334M

A PRACTICAL WAY TO HANDLE 'STICKY' DATA

JIM CARRUTHERS PhD, PEng
CEO NORPAK CORPORATION

www.norpak.ca

NORPAK - WHO??

- Developer of the TV Data Broadcast concepts and standards. Interactive TV again, and again
- The leader in TV Data Broadcasting kit - believe 90% of the world market. 22 years experience
- Over 3,500 NTSC/ PAL/ SECAM, 525/625 line systems in 42 countries. All major data formats. Analogue, serial digital, HD serial digital and MPEG2 video formats
- Encoders, receivers, bridges, monitoring and control

WHAT DATA?

Lots of data terms – VBI, VANC, NABTS, ATVEF, A90, DSMCC, metadata, data essence .... and growing. For purposes of simplicity lets talk about two types:

- 'LAST CHANCE' data. This is data which gets put in at the time of emission. Data broadcast unrelated to the program video is a good example
- 'STICKY' data. A term I have been using to get across the idea that some data needs to be tightly bound to the video

SOME DATA

- Needs to be synchronized with the video and sound – captions, iTV triggers, or
- Not synchronized, but needs to stay with the video and sound, and
- Needs to stand the ‘test of time’ and ‘Murphy’s law’

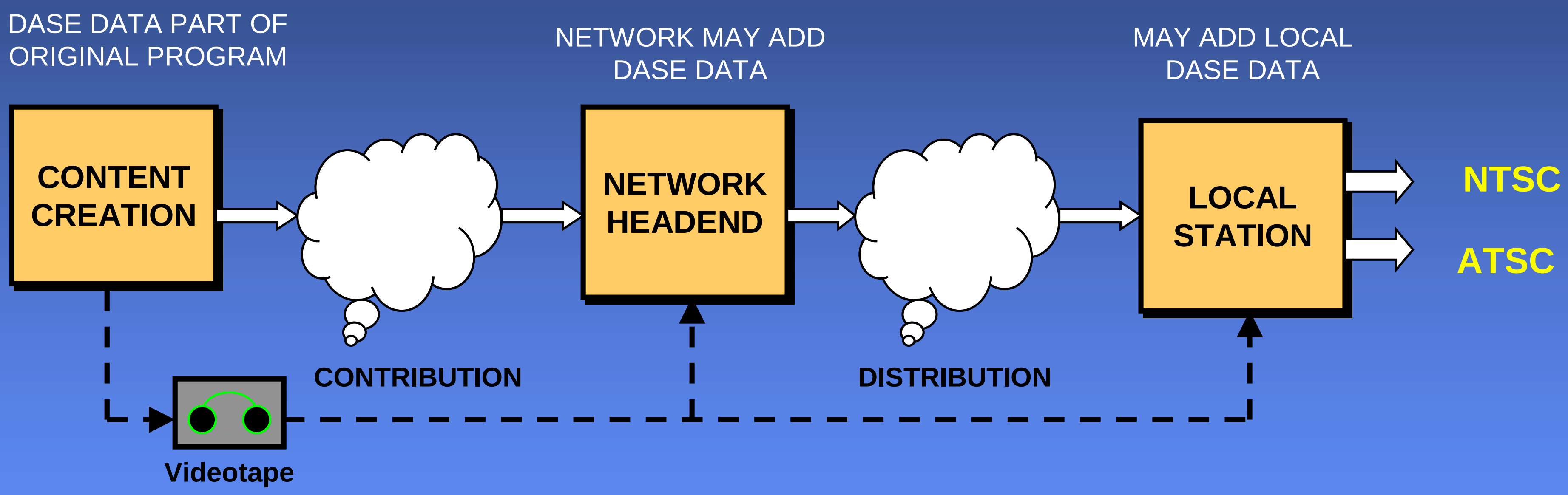
'STICKY' DATA

- Most solutions are focussed on adding data at the point of emission. Fine for data unrelated to the V/A
- 'Sticky' data is content that needs to be bound to the V/A as it travels through multiple plants on its way from the creator to broadcast
- Needs to be transparently stored and retrieved without the complications of locating it on a server somewhere. Delays of seconds, hours, days are commonplace - and what about years down the road?
- Compelling DASE content will originate as an integral part of the creative process – will need to 'stick'

FORWARD / BACKWARD COMPATIBILITY

- Whether the signal is analogue, serial digital or HDTV the 'sticky' data interface should be identical
- Applications such as captioning, VChip, ATVEF and DASE should interface to the data encoder in the same way regardless of the TV standard used
- Makes the move from analogue to digital or HDTV simple and straightforward

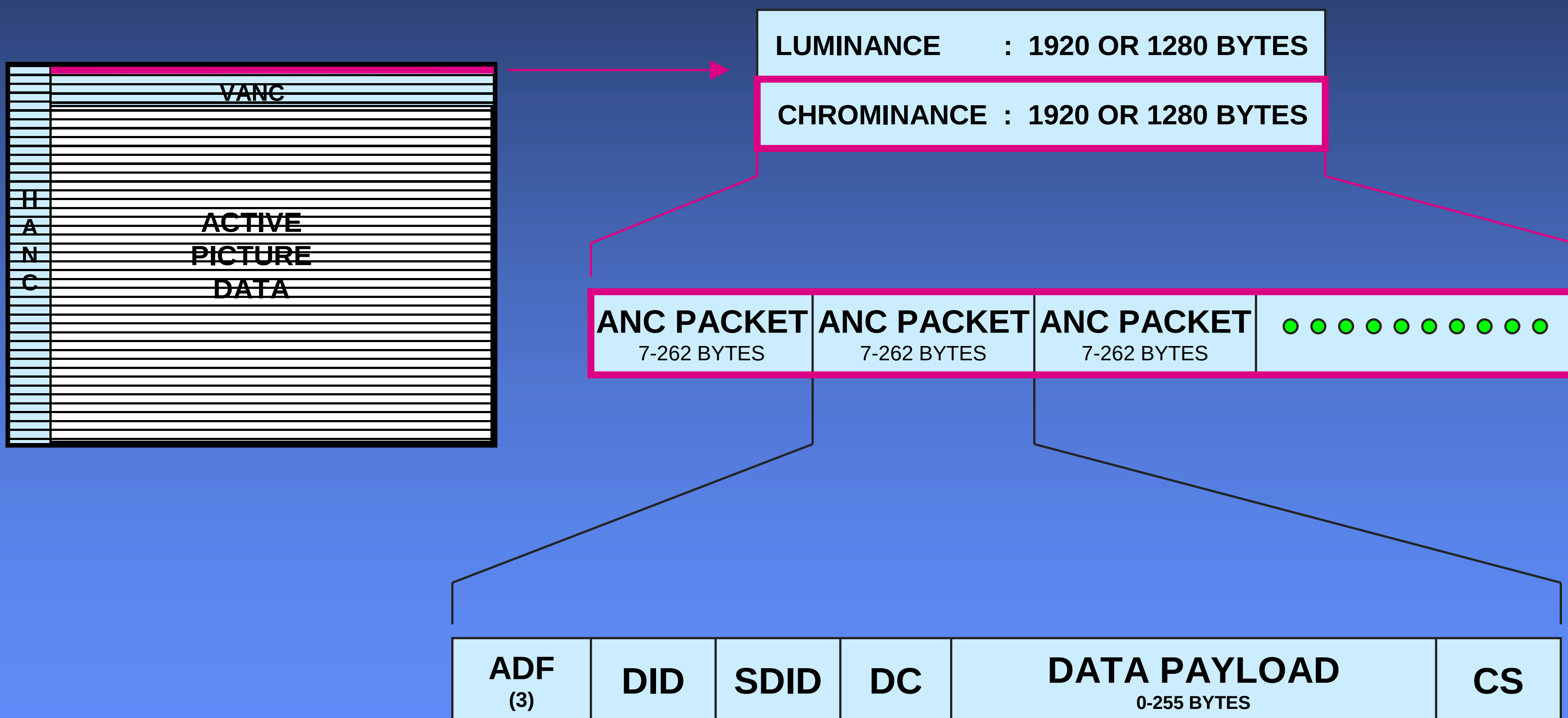
DATA DELIVERY ARCHITECTURE



SMPTE 334M

- The standard for adding data to the SMPTE 292M signal using proven broadcast procedures and processes
- VANC in the SMPTE 292M stream means that the combined V/S/D can be switched, routed, and stored using familiar processes
- Captioning, VChip, ATVEF and DASE interactive data injection continue to be done at the same point in the process, using the same tools and in the same way as it presently is done

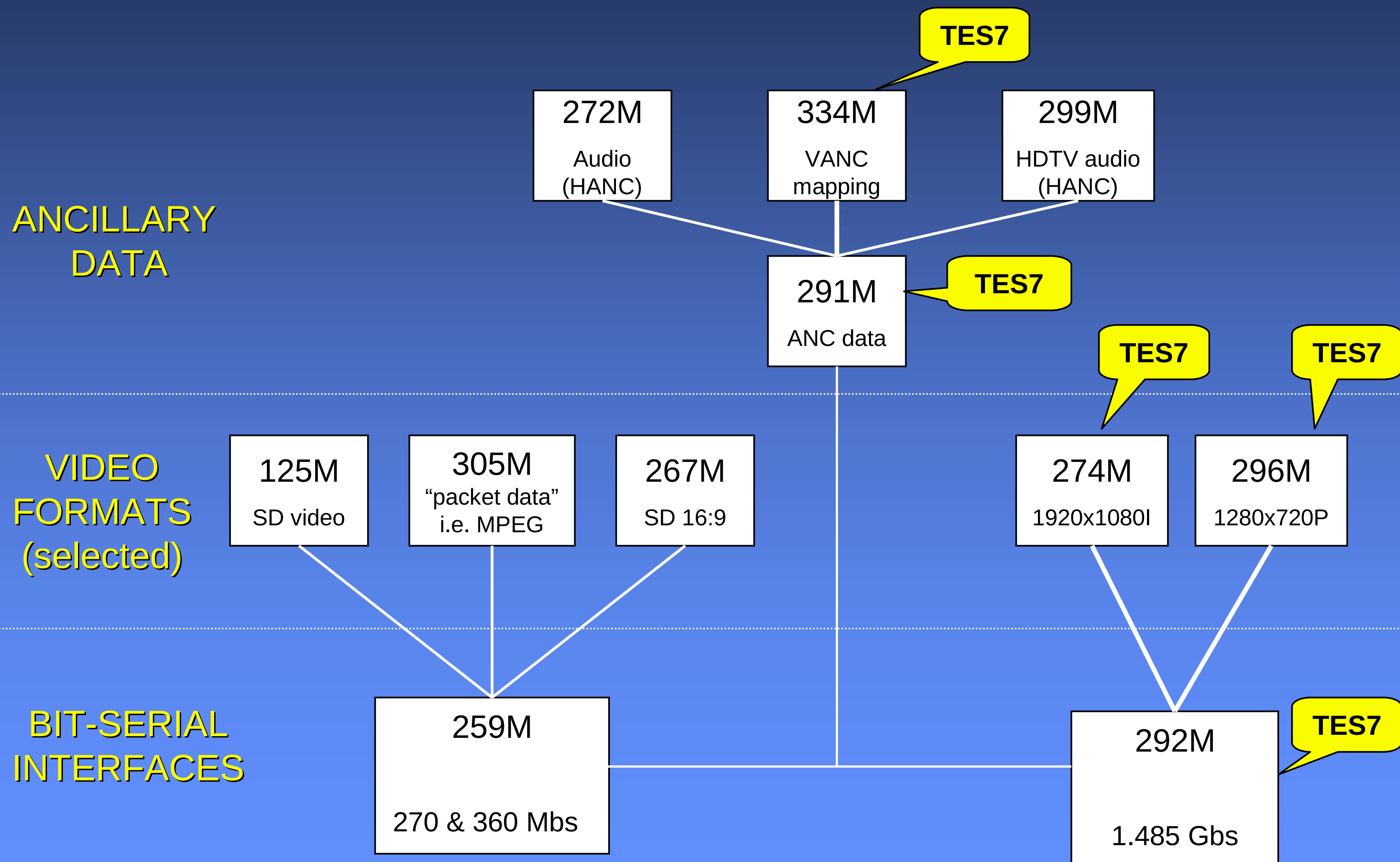
ANCILLARY DATA FORMAT



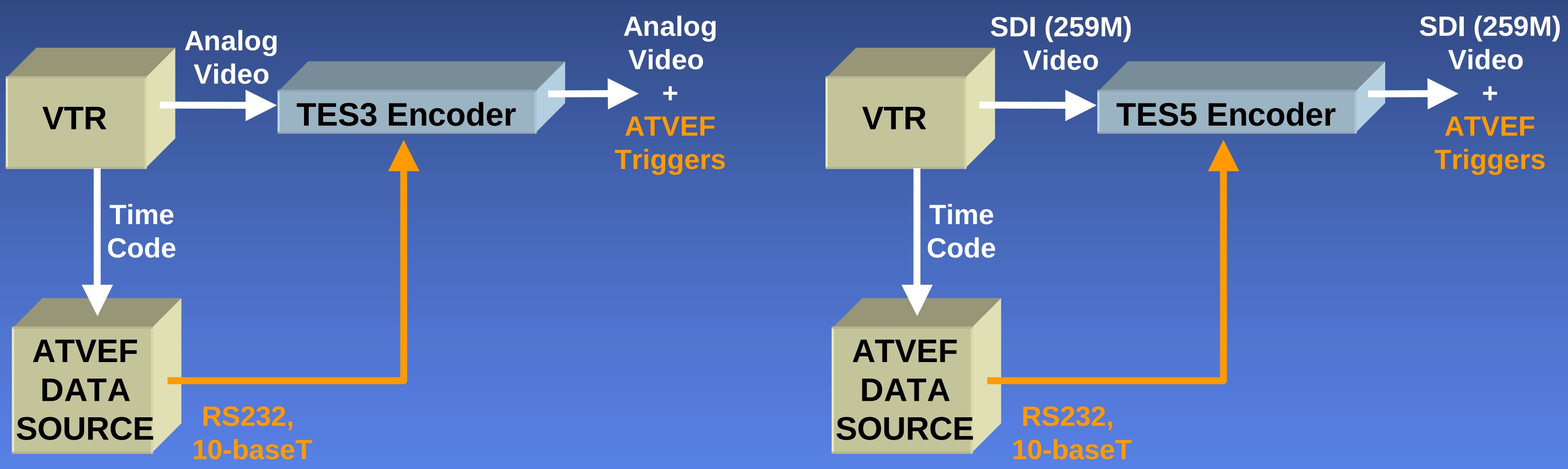
VANC THROUGHPUT

- EACH LINE OF LUMA OR CHROMA CARRIES:
  - 1920 X 60 = 115.2K BYTES/SEC (274M)
  - 1280 X 60 = 76.8K BYTES/SEC (296M)
- EACH LINE CARRIES:
  - 115.2K X 2 = 230.4K BYTES/SEC (274M)
  - 76.8K X 2 = 153.6K BYTES/SEC (296M)
- VANC SPACE CONTAINS:
  - 274M : LINES 1-20, MINUS 2 SWITCHING = 18 LINES
  - 296M : LINES 1-25, MINUS 2 SWITCHING = 23 LINES
- THIS YIELDS:
  - 274M: 33.18 MBS
  - 296M: 25.27 MBS

ENCODER STANDARDS



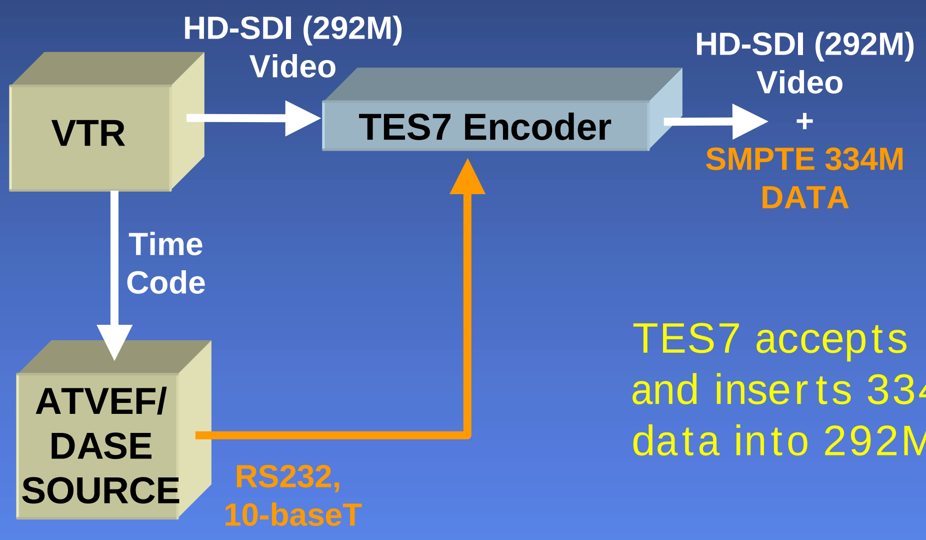
TYPICAL DATA ENCODING OPERATION



ANALOG VIDEO (E.G. NTSC)

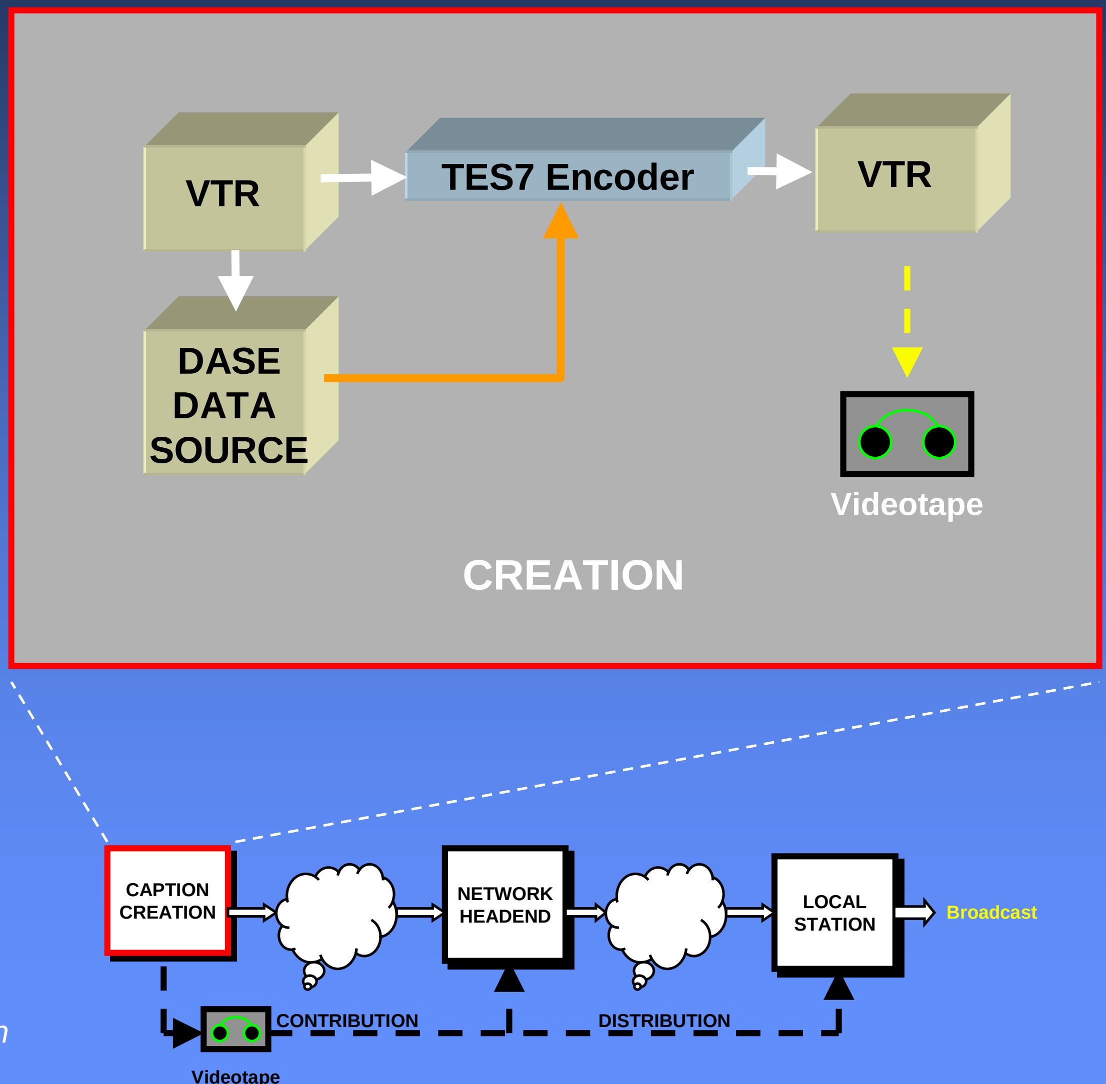
SERIAL DIGITAL (SDI) VIDEO (SMPTE 259M)

SMPTE 334M DATA ENCODING

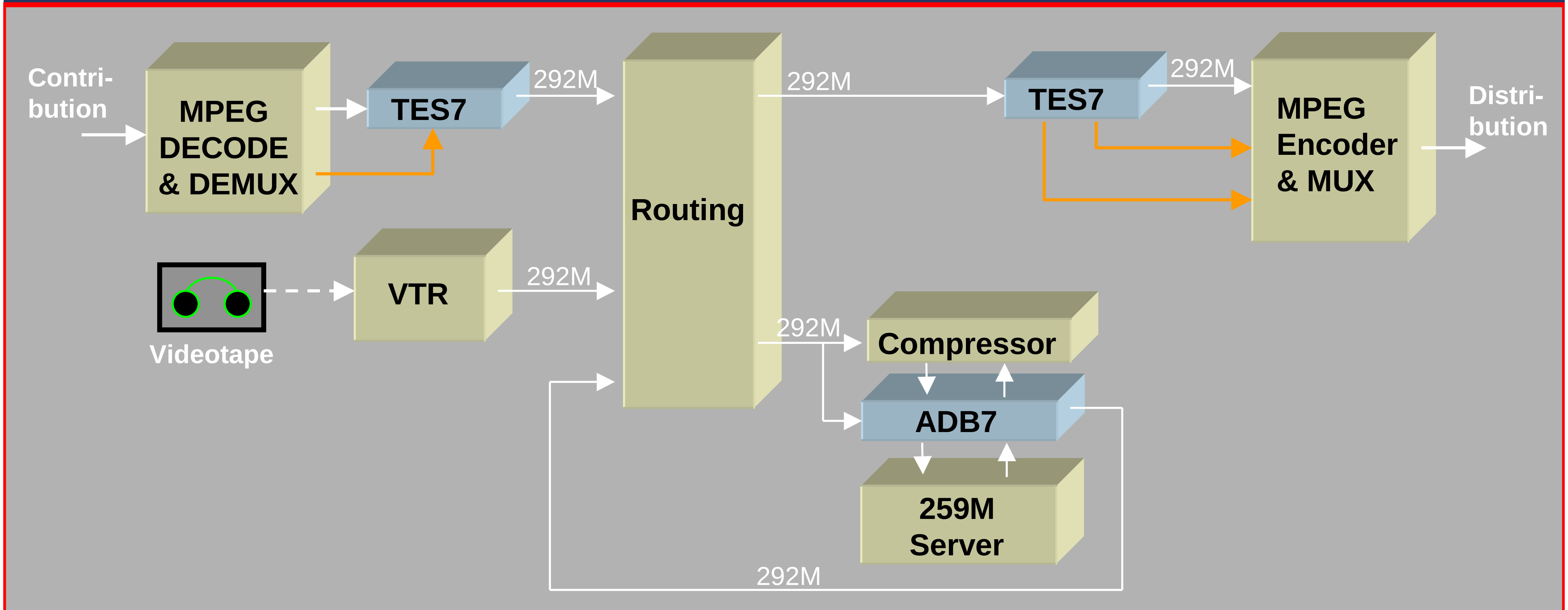


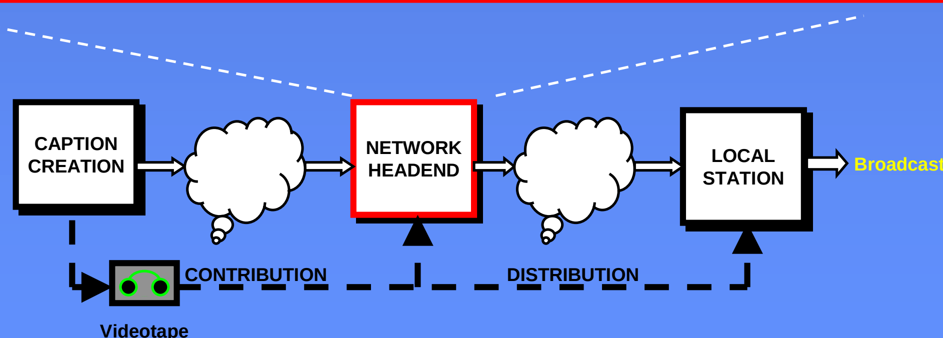
TES7 accepts commands, and inserts 334M VANC data into 292M

DATA DELIVERY (CONTENT CREATION)

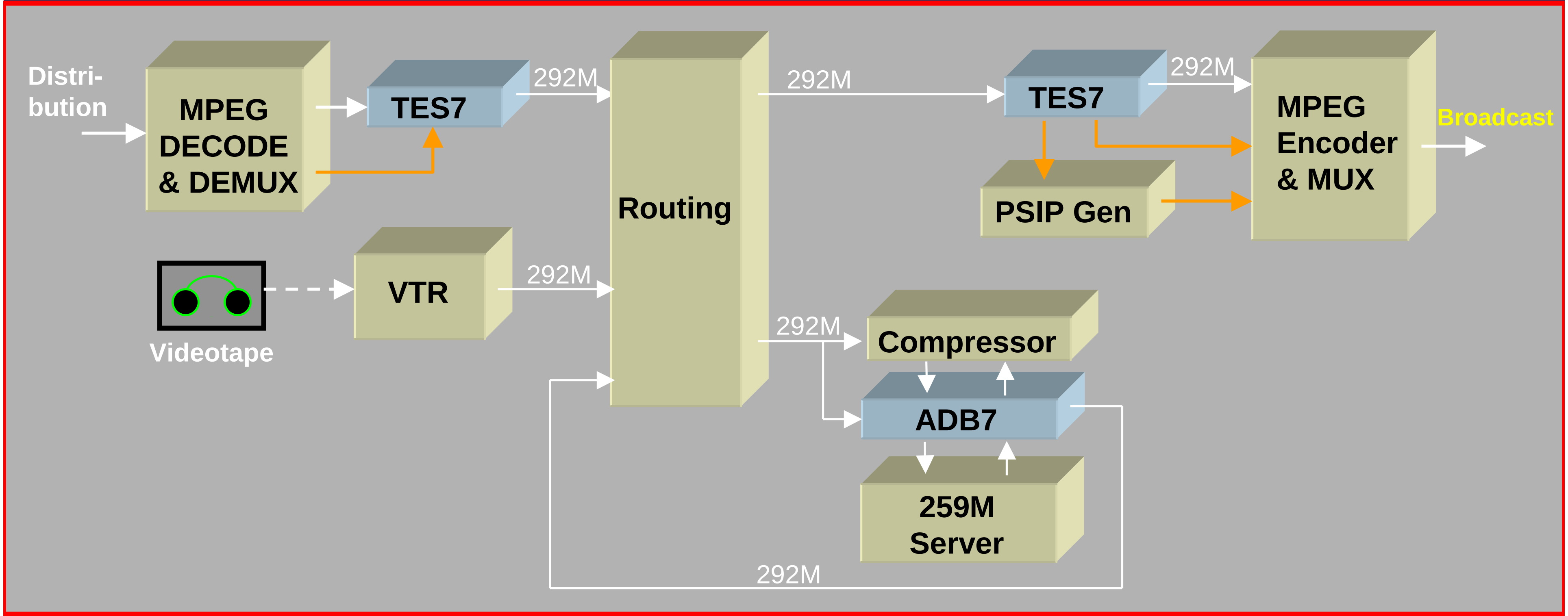


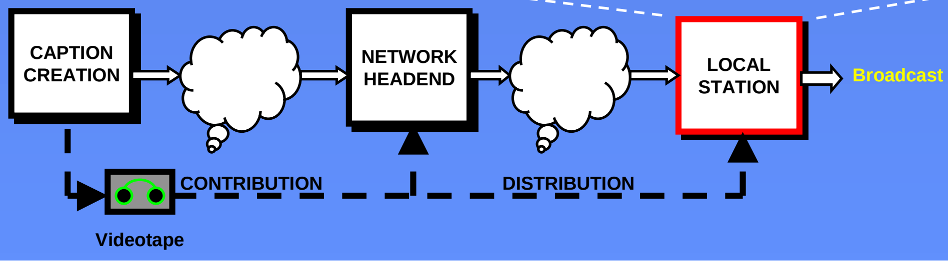
DATA DELIVERY (NETWORK)



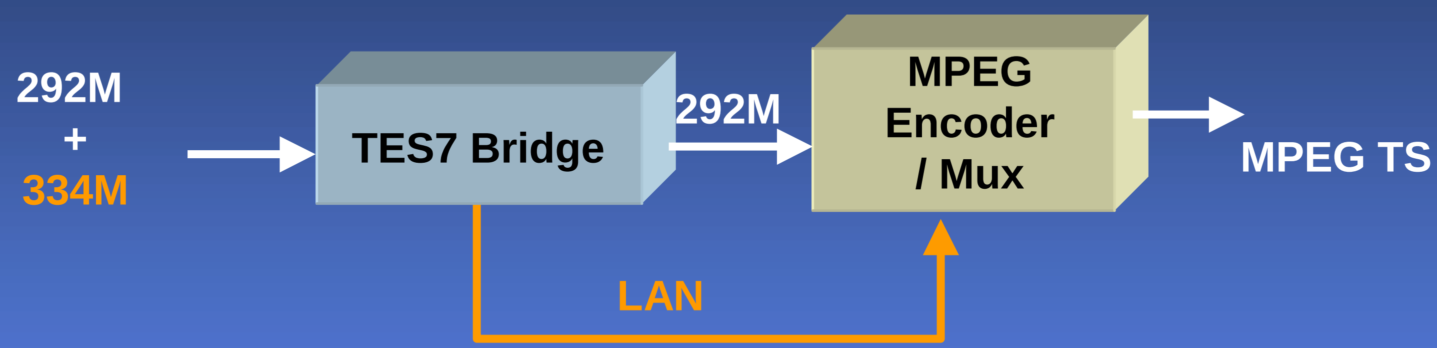


DATA DELIVERY (STATION)



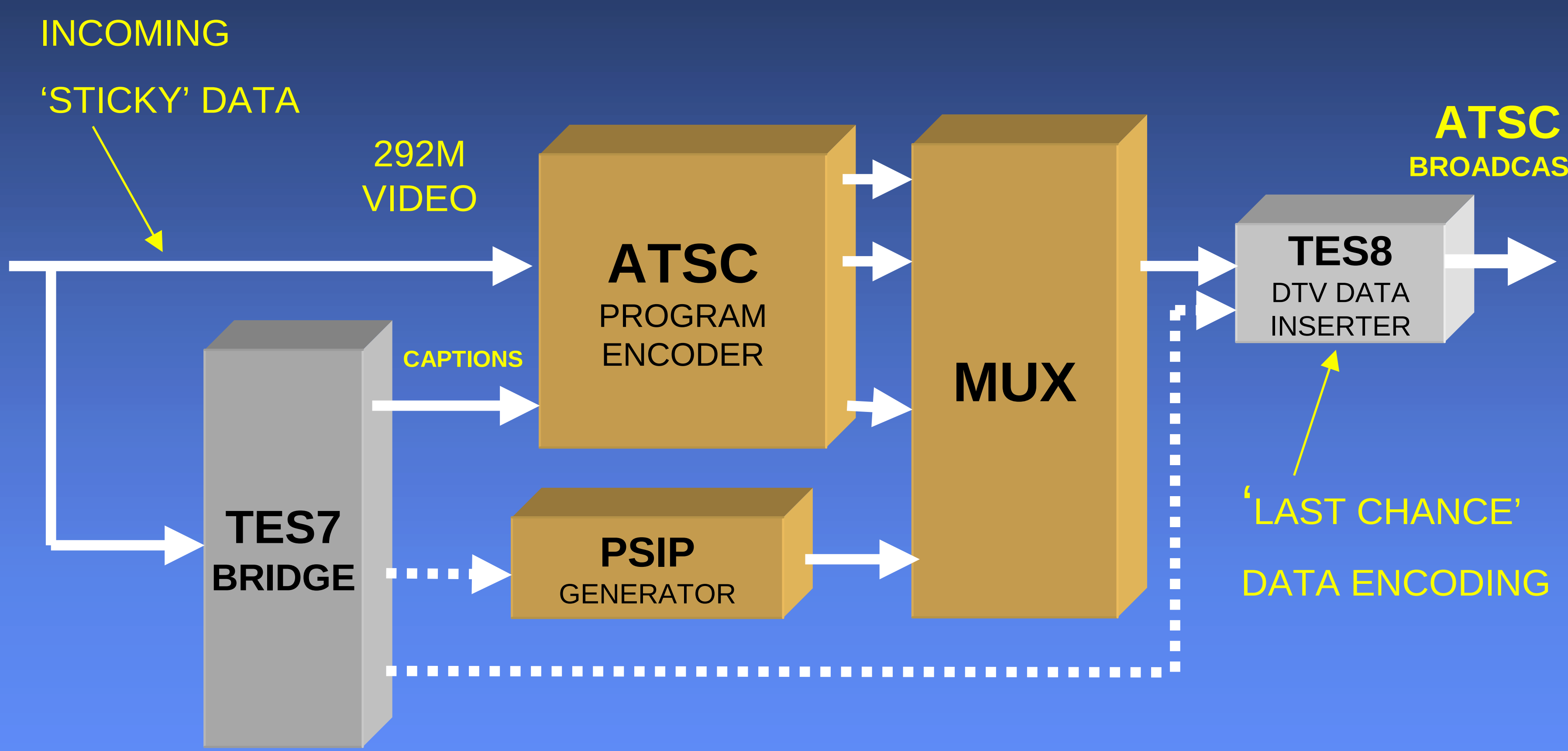


A MISSING (292M-MPEG) LINK



TES7 extracts 334M VANC data and produces data for an MPEG encoder

EMISSION AREA OF A DTV STATION



SUMMARY

- The best place to insert 'last chance' data is at emission, as shown in the previous slide. Many speakers have addressed those concepts.

- 'Sticky stuff'. Compelling content will be born as part of the creative process. Present minimal use of metadata takes off with DTV. Motion picture data, rights data, assets management, etc. It is suggested that SMPTE334M is the way to handle 'sticky' data end-to-end.

Components and Advanced Functionalities of the Next Generation Interactive Programming Guide

Yakov Kamen

iSurtTV,
Sunnyvale, CA
Xakov@isurftv.com

We first describe the major components of the modern interactive programming guide (IPG), its organization and architectural requirements.

We then discuss the need for new schedule and channel management functionalities to simplify channel navigation. In this discussion the following set of advanced IPG functionalities is described: adaptive last channel, single click channel reordering, automatic favorite list generation, non-linear channel scrolling, multiple view modes, fast search without keyboard, channel pause, etc.

Next we propose several possible implementations of IPG components and advanced functions.

At the end several working advanced IPG prototypes are demonstrated.

Advanced Functionalities of the Next Generation Interactive Programming Guide (IPG)

Yakov Kamen, iSurfTV

June 20, 2001

Interactive Programming Guide (IPG) is a software product that allows the user to find necessary information associated with TV Programs transmitted over the broadband network (cable, satellite, ADSL, etc.).

IPG is

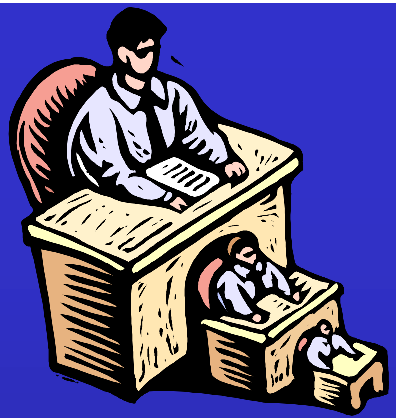
- the most important interactive application for Digital TV;
- the most attractive new source of revenue for IPG broadband network operators
- the major user of STB CPU power and memory
- the most lousily specified interactive application

What does the user want from the IPG?

A. **Preview capabilities.** Get info about the content on the screen without leaving the channel (sneak preview or banner), if the content is interesting to him.

B. **Simple and Fast Scroll.** Find the best program to watch now, if what is playing on the current channel is not what the user wants.

C. **Easy Search.** If there is nothing (or not enough) to watch now, the user wants to search for something available at a later time.



What does the broadband network provider want from the IPG?

A. **Promotion of his service.** Constantly remind the user who is the service provider

B. **Additional Source of Revenue.** To use the guide as an advertisement holder .

C. **Minimal Tech support.** IPG service is free (today) for the user, and nobody wants to support a free service.

D. **Differentiation.** The IPG has to be cool and has to differentiate the provider's service.

IPG Classification

Paper Guide

Passive Programming Guide (PPG)

- Developed for analog TV

Interactive Programming Guide (IPG)

- Supports schedule-related manipulations for digital TV

Advanced Interactive Programming Guide (AIPG)

- IPG which supports one or more of the following advanced functions:
  - PVR
  - Internet
  - Interactive Applications (weather, sport, games, shopping, banking, etc.)

Major Components of IPG

Preview (Mini IPG)

Allows fast sneak information preview

Scroll

Allows fast search of the best currently available show

Search

Allows advanced search of the data in the TV schedule

Remote Controller (RC)

Menu, Settings and Help

Allows to choose one of the components, to set service settings, and find help

Major Components of Advanced IPG

Preview (Mini IPG)

Allows fast sneak information preview

Scroll

Allows fast search of the best currently available show

Search

Allows advanced search of the data in the TV schedule

Menu, Settings and Help

Allows to choose one of the components, to set service settings, and find help

Portal

Allows multiple application navigation

My Shows

Creates listing of recorded shows

Recording Functionality

Provides all conventional recording functions

Remote Controller (RC) and Keyboard

Preview (Mini IPG)-1

Existing Functionality of Mini IPG

Show current event description
Show detailed description (advanced IPG)
Tune to the channel
Automatically disappear in 5- 10 sec
Translucent or opaque implementation

Problems

Only shows current event description
Can be either only translucent or only opaque
Does not allow browsing of other channels without re-tuning
Cannot show advertisements
Unable to support PIP as a part of preview
Does not display the current time

Preview (Mini IPG)- 2

Advanced Functionality of Mini IPG

Show current event description in multiple modes
Show the next few event descriptions
Capability to browse channels with and without tuning to them
Translucent and opaque implementation
Capability to show advertisement
Capability to show PIP and browse in PIP mode
Show the current time, on demand

Scroll-1

Existing Functionality of Scroll

Shows current information as a listing (grid)
Shows advertisement as a part of the listing
Stays tuned to the channel
Automatically shows highlighted event description
Tunes to a channel using its number

Problems

Shows future event description
Uses a single font size for all users
Orders all channels by number only

Scroll-2

Advanced Functionality of Scroll

- Show only current information in different forms (by channel logos, event descriptions, channel numbers, screen snapshots)
- Adjust font size by interactive zooming
- Shows single event description or a short list of coming events
- Shows a page number
- Sort channels by A-Z, numbers, categories
- Sort by popularity
- Scroll in the “Mosaic” channel

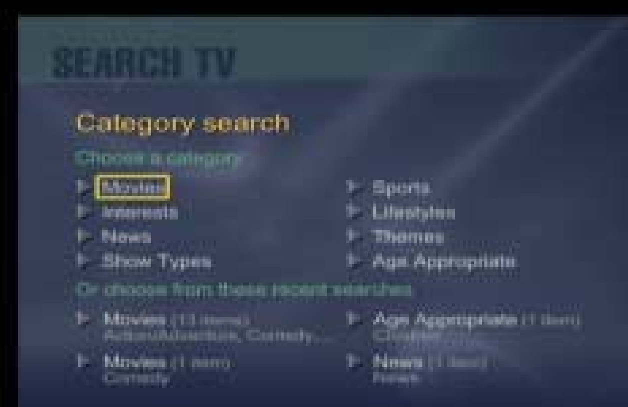
Search-1

Existing Functionality of Search

Uses the grid for a “toggling” search
Allows searching by title, phrase, time, day, channel #
Allows searching by categories and subcategories

Problems

TV program and ads are invisible when in the search mode
Always searches over the entire data set
Can not sort by similarity



Search-2

Advanced Functionality of Search

- Search in categories
- Adjust font size by interactive zooming
- Keeps advertisement and current TV program on the screen
- Shows a page number
- Sort results of the search by channel #, channel names, popularity, etc.
- Visual Search

Remote Control

Problems

No memory function (like in cellular phone)
Single last channel option

Advanced Functionality of Remote Controller

Using keys as memory buttons
Multiple last channels

My Show

Problems

Recorded shows can not be sorted
No bookmarking capabilities

Advanced Functionality

Sorting of shows by A-Z, date recorded, category
Adding notes and bookmarks to recorded shows

SMPTE Declarative Data Essence: Comparison to ATSC DASE

John Barkley, **Michael Dolan,**

NIST Chair, SMPTE DDE
jbarkley@nist.gov miked@tbt.com

Michael Koo, Andrew McCaffrey, Len Gebase, and Murugiah Souppaya

NIST
Gaithersburg, MD 20899

The 70th Anniversary Issue of Business Week recognizes both TV and the Internet as technologies which have had a profound effect on the economy. Interactive TV (ITV) can be characterized as a convergence of TV and the Internet. The effect on the economy of their convergence is expected to be significant.

Neither TV nor the Internet would have been so successful without standards. Likewise, their convergence, ITV, needs standards in order to succeed. !TV standards are needed in the following areas: content. receivers, application initiation/termination, application synchronization with video/audio broadcast, and content delivery.

The Declarative Data Essence (DDE) Ad-hoc Group of the D27 Technical Committee of the Society for Motion Picture and Television Engineers (SMPTE) is developing ITV standards that provide basic functionality for ITV embodying current practice, known collectively as DDE-l. The Advanced Television Enhancement Forum (ATVEF) specification is the basis for the DDE-l effort. The T3/S17 Specialist Group within the Advanced Television Standards Committee (ATSC) is developing the DTV Application Software Environment (DASE) standard for the next generation of ITV.

This paper summarizes the SMPTE DDE-l specifications and compares DDE-l to DASE. In addition, this paper describes the functionality specified within DDE-l and DASE that enable content developers to present and manage ITV applications. This paper compares DDE-l and DASE with regard to: content types, application initiation/termination, synchronization, and content delivery.

Content developers include producers of TV programs and commercials. TV programming consists of entertainment segments interspersed with commercials. Entertainment and commercial segments are usually produced independently from each other. ITV Content developers need to be able to develop content that runs on all receivers, and that does not interfere with content developed independently by others. Both DDE-l and DASE specify functionality needed for sequencing ITV applications synchronized with a TV broadcast. This paper compares DDE-l applications (called "enhancements") and DASE applications, and gives examples of ITV applications.

SMPTE Declarative Data Essence: Comparison to ATSC DASE

John Barkley, NIST jbarkley@nist.gov

Michael Dolan, Chair, SMPTE DDE miked@tbt.com

Michael Koo, Andrew McCaffrey, Len Gebase, and Murugiah Souppaya

NIST
Gaithersburg, MD 20899

1 Introduction

Interactive TV (ITV) can be characterized as a convergence of TV and the Internet. Both TV and the Internet, as separate technologies, have already had a profound effect on the economy [BW70]. The effect on the economy of their convergence is expected to be significant.

Neither TV nor the Internet would have been so successful without standards. Likewise, their convergence, ITV, needs standards in order to succeed. ITV standards are needed in the following areas:

- Content: standards specifying ITV content types. Content producers must know which content types to use in developing ITV applications so that the creation and distribution packages are interoperable.
- Receivers: standards specifying receiver behavior so that content displays the same on all receivers.
- Application Initiation/Termination: standards specifying how ITV applications are initiated and terminated. Content producers must know how to start/stop their applications, and how to ensure that applications do not interfere with each other.
- Synchronization: standards specifying how content is synchronized with the video/audio broadcast. Content producers must be able to synchronize their ITV applications to the broadcast video/audio so that the same interactive experience is provided to the viewer on all receivers.
- Delivery: standards specifying how content is delivered to the receiver.

The Declarative Data Essence (DDE) Ad-hoc Group of the D27 Technical Committee of the Society for Motion Picture and Television Engineers [SMPTE] is developing ITV standards that provide basic functionality for ITV embodying current practice, known collectively as DDE-1. The Advanced Television Enhancement Forum [ATVEF] specification is the basis for the DDE-1 effort. The T3/S17 Specialist Group within the Advanced Television Standards Committee [ATSC] is developing the DTV Application Software Environment (DASE) standard for the next generation of ITV.

This paper[1] summarizes the SMPTE DDE-1 specifications and compares DDE-1 to DASE.[2] In addition, this paper describes the functionality specified within DDE-1 and DASE that enable content developers to present and manage ITV applications. This paper compares DDE-1 and DASE with regard to: content types, application initiation/termination, synchronization, and content delivery.

Content developers include producers of TV programs and commercials. TV programming consists of entertainment segments interspersed with commercials. Entertainment and commercial segments are usually produced independently from each other. ITV Content developers need to be able to develop content that runs on all receivers, and that does not interfere with content developed independently by others. Both DDE-1 and DASE specify functionality needed for sequencing ITV applications synchronized with a TV broadcast. This paper compares DDE-1 applications (called "enhancements") and DASE applications, and gives examples of ITV applications.

2 DDE-1 Compared to DASE: Overview

Table 1 provides a summary comparison between DDE-1 and DASE.[3] DDE-1 consists of seven specifications, three for content ([DDE1], [DDE1-DOM0], [DDE1-lid]), and four for binding content to broadcast streams ([DDE1-UHTTP], [DDE1-IPM], [DDE1-NTSC], [DDE1-PAL]). DASE consists of eight specifications ([DASE], [DASE-DA], [DASE-PA], [DASE-API], [DASE-FONT], [DASE-SEC],[4] [DASE-CONF], [ATSC-ARM]).

In DASE, ITV applications are characterized as either declarative applications (DA), whose content types are primarily Web pages, or procedural applications (PA), whose content types are primarily Java. More specifically, in DASE, a DA is a collection of resources whose "root," as identified by ATSC A/90 [ATSC-A90] application signaling [ATSC-ARM] (see section 6) is of type application/xdml+xml. A PA is a collection of resources whose root is of type application/javatv-xlet.

DDE-1 specifies a DA type application. More specifically, DDE-1 applications, commonly called an "enhancements," are analogous to DASE Purely Declarative Applications (PDA). PDAs are DASE DAs where every resource has a declarative content type.

[1] Because of the nature of this paper, it is necessary to mention vendors and commercial products. The presence or absence of a particular trade name product does not imply criticism or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available.

[2] The descriptions of DDE-1 and DASE functionality in this paper are based on the DDE-1 and DASE specifications available at the time of publication. Some of these specifications may undergo revision subsequent to this paper's publication. Moreover, the descriptions of DDE-1 and DASE functionality in this paper are summaries. For details, see the specifications.

[3] In this paper, features described as "specified" within DDE-1 or DASE means that behavior semantics for these features are specified within the DDE-1 or DASE standards. If features are identified as "not specified," content developers should be aware that such features might not be supported in a particular implementation of a standard.

[4] DASE Security is very much a work in progress and is not described in this paper. DDE-1 does not specify security features.

	DDE-1	DASE[5]
DA Content Types	text/html, text/plain, text/css, text/ecmascript, application/tve-trigger, image/png, image/jpg, audio/basic	application/xdml+xml, text/css, text/ecmascript, DASE Common Facilities[6]
PA Content Types	not specified	application/java, application/java-xlet, application/octet-stream, DASE Common Facilities[6]
Application Initiation/Termination	triggers/triggers	ATSC A/90 application signaling/triggers
Synchronization	triggers	triggers
Content Delivery	IP Multicast over NTSC, PAL/SECAM, ATSC and DVB; or backchannel	Broadcast using a subset of ATSC A/90

Table 1: DDE-1 vs. DASE

Receivers capable of running DDE-1 enhancements have two basic design concepts:

- Embedded within the receiver is Web browser software. The broadcast video and audio can be referenced within the HTML document as the URL "tv:".
- The two media, TV and the Internet, are synchronized by embedding "triggers" within the broadcast. Triggers are content which upon receipt, signal enhancements to be initiated/terminated, or ECMAScript code to be executed within an enhancement already running.

Every resource in a DDE-1 enhancement or DASE application is one of the content types as shown in table 1. Within the DDE-1 specifications, there is no equivalent to a DASE PA. For more information on differences in content types between DDE-1 and DASE, see section 3.

Both DDE-1 and DASE specify functionality for the initiation and termination of an application. In DDE-1, enhancements are initiated/terminated by triggers. In DASE, applications are initiated by ATSC A/90 application signaling, and terminated by triggers. See Section 4 for more information.

Both DDE-1 and DASE specify functionality, by means of triggers, for synchronizing running ITV applications with the video/audio. However, DDE-1 and DASE triggers have a different format. See section 5 for more information about synchronization.

---

[5] In addition to the DASE content types listed in this table, the DASE Security specification under development defines the additional content types: application/dase-manifest, application/dase-permission, application/dase-signature

[6] DASE Common Facilities consist of the content types: application/dase-trigger, image/jpg, image/png, video/mng, audio/basic, video/mpeg, video/mpv, audio/ac3, application/font-tdpfr, application/jar.

DDE-1 specifies the delivery of interactive content by either the broadcast or by means of a backchannel,[7] typically an Internet connection. DDE-1 specifies a content binding for NTSC and PAL/SECAM analog systems. Additionally, 3rd parties have specified its binding to ATSC and DVB. In all cases, the binding relies on an IP Multicast encapsulation of the data, as defined in DDE-1. DASE specifies interactive content delivery by means of the ATSC A/90 Standard.[8] See section 6 for more information.

3 DDE-1 Content Compared to DASE Content

Table 1 shows the content types specified in DDE-1 and DASE. DDE-1 specifies content currently in widespread use not only in ITV, but also on the Web. DASE specifies content more recently adopted and under development by the [W3C], as well as additional types. In addition, DASE specifies Java content types for PAs, which can also be part of DASE DAs. The following subsections compare DDE-1 and DASE content in the categories of content markup, stylesheet, script, and document object model.

3.1 Markup Content

DDE-1 specifies HTML 4.0 [HTML4]. DASE DA specifies the content type application/xdml+xml which consists of a subset of XHTML 1.0. The markup content for both DDE-1 and DASE DAs are very similar. The content type of the root of a DASE DA must be application/xdml+xml. Although markup content for DASE DAs is specified primarily as an XML DTD, the receiver is not required in DASE to be a validating XML processor, i.e., to be capable of validating content relative to any arbitrary DTD. A DASE receiver is only required to validate markup content according to the DTDs defining the content type application/xdml+xml.

3.2 Stylesheet Content

DDE-1 specifies Cascading Style Sheets level 1 [CSS1]. DASE DA specifies a subset of Cascading Style Sheets level 2 [CSS2]. CSS2 includes the functionality of CSS1, and specifies additional selectors. These include new pseudo-elements, e.g., the first letter of a paragraph, and new pseudo-classes, e.g., a link which has been visited. In addition, CSS2 specifies new types of selectors, such as attribute selectors, which can identify markup elements by the presence of an attribute and/or attribute value, and parent/child selectors which can identify markup elements based on a parent/child relationships.

3.3 Script Content

DDE-1 specifies ECMAScript 2nd Edition [ES2]. DASE DA specifies ECMAScript 3rd Edition [ES3], which includes the functionality of ECMAScript 2nd Edition. In addition, ECMAScript 3rd Edition specifies a "switch" statement, try/catch exception handling, and native regular expression objects. The functionality of these regular expression objects is modeled after regular expressions in Perl 5.

---

[7] DDE-1 informatively defines "backchannel" as "a connection from a receiver to the Internet or back to some server."

[8] Backchannel functionality may be included in the next version of DASE, commonly referred to as "DASE-2."

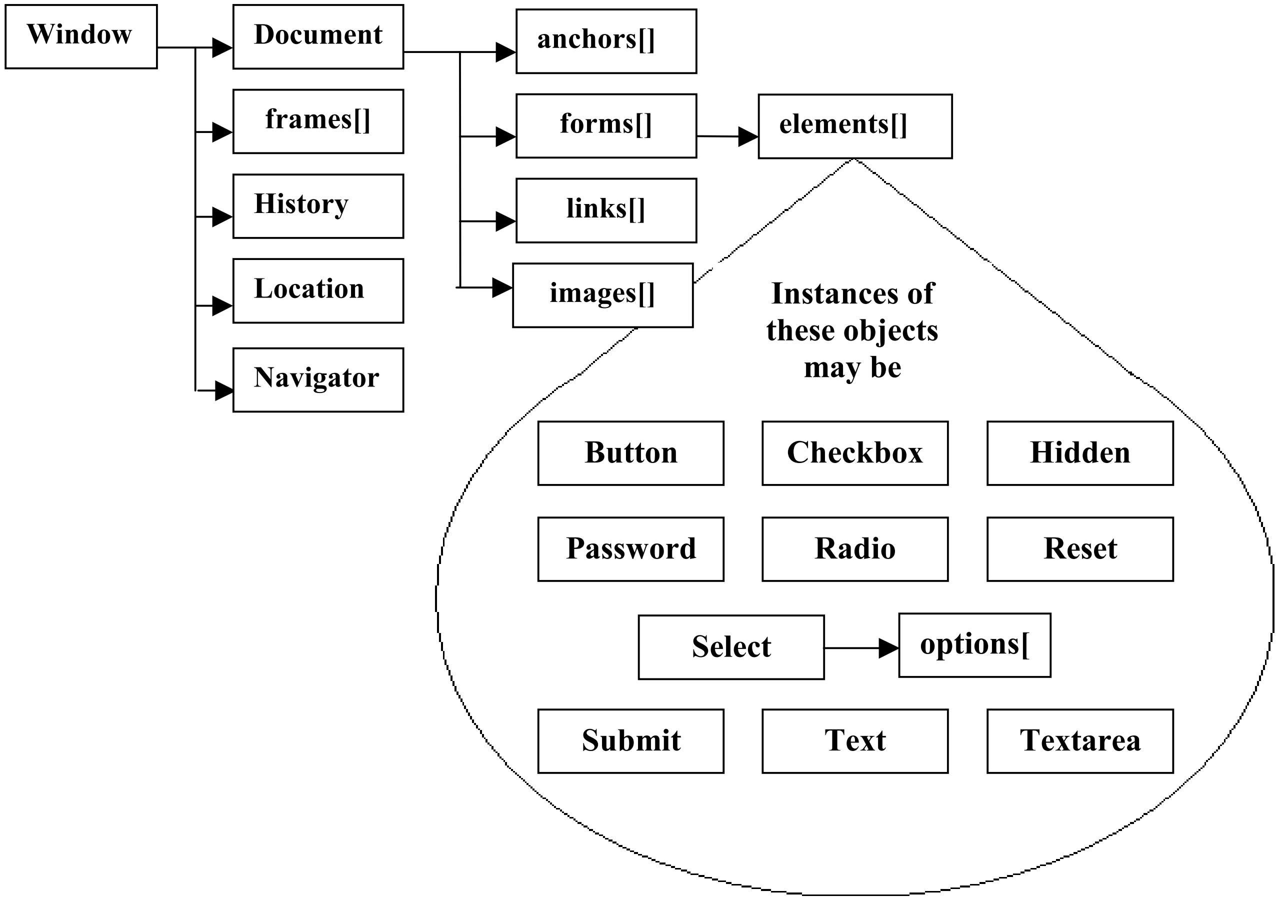


Figure 1: DOM-0 Objects and Their Relationships

3.4 Document Object Model

The document object model for DDE-1 is Document Object Model Level 0 (DOM-0) [DDE1-DOM0]. DOM-0 generally has the functionality of the object model common to both Microsoft Internet Explorer 3.0 and Netscape Navigator 3.0. This functionality is currently in widespread use on the Web today. Figure 1 shows the DOM-0 Objects and their relationships. For these objects, DOM-0 specifies basic functionality for reading/mutating, and processing of events.

DASE DA specifies a subset of the DOM Level 2 (DOM-2) Object Model, specifically, [DOM2-CORE], [DOM2-HTML], [DOM2-VIEWS], [DOM2-STYLE], and [DOM2-EVENTS]. Although the DOM-0 and DOM-2 object models are different, the DOM-2 object model includes most of the functionality of DOM-0 and more. DOM-2 specifies the reading/mutation of almost all elements of an HTML document, in particular scripts and stylesheets. With DOM-2, changing information on a page can be accomplished by mutation, rather than by replacing an entire frame. This is particularly useful when the amount of information to be changed is small, e.g., a number in a table.

The DOM-0 event model is based on intrinsic events as specified in HTML 4 syntax. In contrast, the DOM-2 event model is more robust and its behavior more well defined. While not technically part of the DOM, both DDE-1 and DASE specify native objects in support of current practice. These are the “Window,” “Location,” “History,” and “Navigator” objects.

	DDE-1 Enhancement	DASE Purely Declarative Application
Producer Initiated	trigger with name and URL different from current application; instantiation automatically or with viewer selection ([DDE1], 4.4, 4.6.1, 5.3)	ATSC A/90 application signaling; instantiation automatically or with viewer selection ([DASE], 5.1.2.2); [ATSC-ARM], 5.4)
Viewer Initiated	not specified	viewer navigation to content type application/xdml+xml ([DASE-DA], 5.1.1.6.1.1.1)
Producer Terminated	trigger with script that initiates a navigation to tv: ([DDE1], 4.6.1)[9]	application root entity top level window object executes window::close() ([DASE-DA], 5.3.1.2.9.4.3)
Viewer Terminated	viewer navigation to tv: ([DDE1], 4.6.1)[9]	viewer navigation to content types video/mpeg ([DASE-DA], 5.1.1.6.1.1.2) or application/xdml+xml, ([DASE-DA], 5.1.1.6.1.1.1)

Table 2: ITV Application Initiation/Termination

4 Application Initiation/Termination

TV programming usually consists of entertainment interspersed with commercials. Content developers for ITV include producers of both kinds of content. Entertainment and commercial segments are usually produced independently. ITV content developers need to be able to develop content that runs on all receivers, and that does not interfere with content developed by others independently.

DDE-1 specifies a simple ITV application initiation/termination capability, while DASE specifies a more complex capability of initiation, termination, and suspension (see [ATSC-ARM], section 5). For example, in DASE, when a viewer changes channel, a DASE application is suspended, and may resume under certain circumstances when the viewer returns to that channel.

Table 2 summarizes the functionality within DDE-1 and DASE for initiating and terminating an application. The term "producer" refers to the content developer and the term "viewer" refers to the end-user interacting with the content. Within DASE, there is functionality for the producer to craft the application in such a manner so that the viewer can be granted the option for initiating/terminating an application. Within DDE-1, only the producer can initiate an enhancement.

---

[9] Note that, depending on the implementation, a trigger for a new enhancement that arrives before the termination of the current enhancement may also terminate an application.

Both DDE-1 and DASE specify receivers minimally capable of running only one DA at a time. Because entertainment and commercial segments of a broadcast are interspersed, producers of ITV applications need to be able to initiate content for their segment, and then, at the end of the segment, terminate their content in order to allow the next segment the opportunity to run. For producers, two approaches to accomplishing this are as follows.

The first approach is for segment producers to develop a DA that establishes control of the receiver at the beginning of their segment, and releases this control at the end of the segment so that the application for the next segment can run (see table 2). In this approach, a DDE-1 enhancement (an HTML document) or a DASE PDA (an XDML document) is instantiated and displayed by the receiver. In DDE-1, this is accomplished by the producer creating an appropriate SDP (Session Description Protocol) record if Transport B (see section 6) and sending a trigger referencing the URL of the new enhancement. In DASE, the producer uses ATSC A/90 application signaling to initiate a new PDA. In DASE, the producer can also provide the viewer a link within the currently running application to initiate a new application.

When a segment ends, DDE-1 enhancements can be terminated by a trigger that navigates to the URL "tv:" from the topmost page of the enhancement, or by a producer provided viewer selection within the currently running application that navigates to "tv:" from the topmost page. Navigation to "tv:" from the topmost page also displays full screen video.

Termination of a DASE DA can be accomplished by a trigger that results in the execution of a window::close() within the root top level window object, or by a viewer selection within the currently running application which causes the application root top level window object to execute window::close(). In addition, viewer navigation from the root top level window object to a root entity, which is either content of the type video/mpeg or application/xdml+xml, terminates the current DA. If the content type is video/mpeg, then full screen video is displayed. If the content type is application/xdml+xml, then that content becomes the new DA.

The second approach is for producers of several sequential segments to share the topmost page of an application. For this approach, content for each segment is presented by means of frames subordinate to the shared single topmost page, which serves as an "executive" for frames associated with the segments. The first approach, described above, is used for initiation/termination of the topmost page of the application. Both DDE-1 and DASE specify functionality which allows an application to be present in the background while the viewer sees the video full screen, as though there were no application running. This is a particularly useful feature for this second approach. The viewer is given the choice of opting out of the interactive experience while the application continues in the background, ready to receive the frames for the next segment.

5 Synchronization

In DDE-1, triggers consists of a required URL, the required "tve" attribute which identifies the version of DDE-1 content to which the trigger conforms, optional name/value attributes ("name", "expires", "script"), and depending on the means of delivery, a checksum. The URL of the trigger identifies the topmost page, i.e., the root in DASE terminology, of the application. If the URL equals the URL of the root of the currently running application, then the trigger signals the running application; otherwise, the trigger may initiate a new application and must have a "name" attribute to be valid. The "expires" attribute indicates the time after which the trigger is no longer valid. The "script" attribute contains the script text to be executed upon receipt of the trigger. DDE-1 triggers deliver a single signal, i.e., a single script. The target of the script is the trigger's URL, i.e., the root of the application. For details, see [DDE-1], appendix E.

In DASE, triggers are the content type application/dase-trigger. Like all DASE content, triggers are delivered by means of the ATSC A/90 framework (see section 6). The DASE trigger mechanism is built upon the functionality of DOM-2 Events [DOM2-EVENTS]. A DASE trigger consists of several events of type "script", i.e., a single DASE trigger can deliver several signals to an application. The target of the script event can be any URI identifying declarative content within the application, and can make use to the "bubbles" and "cancelable" features of DOM-2 Events. Each script event is invoked by means of the DOM-2 Events infrastructure. For details, see [DASE-DA], section 4.5.

6 DDE-1 Content Delivery Compared to DASE

DDE-1 specifies two types of interactive content delivery: Transport A, and Transport B. With Transport A, only triggers are delivered with the broadcast stream. Other primary content is actively acquired by the receiver by means of a backchannel, usually an Internet connection. DDE-1 specifies an interface that enables an enhancement to obtain information about the availability of a backchannel and whether it is currently connected. The URLs in DDE-1 Transport A triggers identify the content to be fetched over the backchannel using HTTP Version 1.1.

With Transport B, all content, including triggers, are delivered by means of the broadcast stream. A backchannel may optionally be present for transactions or general Web browsing, but content for applications is delivered via the broadcast. DDE-1 Unidirectional Hypertext Transport Protocol (UHTTP) [DDE1-UHTTP] is the protocol used for content delivery in Transport B. The binding of application announcement metadata and UHTTP to IP Multicast is specified in DDE IP Multicast Encapsulation [DDE1-IPM]. The binding of DDE-1 content for both Transport A and Transport B to the NTSC video standard is specified in DDE-1 NTSC IP and Trigger Binding to VBI [DDE1-NTSC]. The binding of DDE-1 content for both Transport A and Transport B to the PAL/SECAM system is specified in DDE-1 TRIGGERS and IP Binding to PAL/SECAM SYSTEM [DDE1-PAL].

In DASE, all content is delivered in the broadcast stream according to ATSC A/90 [ATSC-A90] and the Application Reference Model [ATSC-ARM]. Three types of information about an application are delivered: announcement, signaling, and content. Application announcement information indicates the future availability of an application, and consists of metadata about the application, such as, name, description, availability time and duration. Application signaling information indicates the imminent arrival of an application's content, and includes the identification of the application's root entity. Following application announcement and signaling information is the application's content. Each content element is delivered with its identifier and content type.

ATSC, in addition to more native generic resource carriage in A/90, also includes a specification for IP Multicast binding [ATSC-IPM]. Thus, with the A/90 and DDE-1 IP Multicast specifications, there is a standard way of broadcasting DDE-1 content with A/90.

For more information about data broadcasting applications, specifically within the ATSC data framework, see [ATSC-DATA].

7 Example ITV Applications

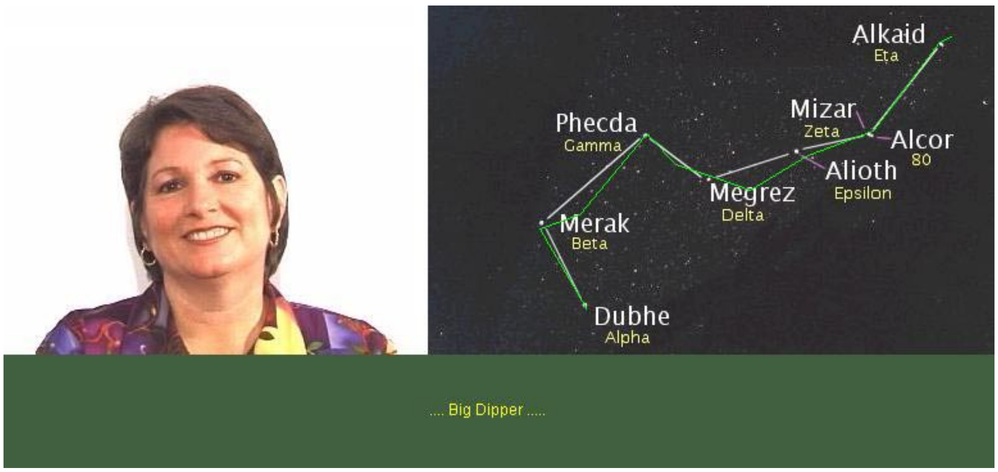


Figure 2: Educational ITV Application

ITV applications range in complexity from the very simple to the very sophisticated. An example of a simple ITV application is one which offers the viewer a pizza delivery. This application could be synchronized with the broadcast of a sporting event or a commercial for the pizza retailer. This example illustrates a simple e-commerce application. It is an example of an ITV application designed for the functionality of the backchannel informatively defined in DDE-1. The backchannel, an Internet connection or possibly some form of proprietary phone connection, is the means of completing the real-time transaction implied in this example.

ITV applications can also provide a richer viewing experience. One such example is an application that provides the viewer the opportunity to interactively obtain additional information about a sporting event during the broadcast. Consider the broadcast of a golf tournament. The viewer is offered the opportunity to browse information about the score for the current player, the details of the current green, player statistics, and information about the tour. This application is an example of an application that can be designed for the functionality of either DDE-1 or DASE. This application requires neither backchannel nor high performance procedural environment.

Figure 2 shows a sophisticated ITV application. It is an educational program for learning about constellations. Developed at NIST, it is an example of an ITV application designed for the functionality specified in DASE. In the application, a narrator first describes several constellations while the outline of the constellation is shown against the star field in the night sky. Then, the viewer is shown a picture of a constellation's star field, and encouraged to draw the outline of the constellation on the star field shown. The current outline is then displayed along with the viewer's drawing so that the viewer can compare. Figure 2 shows the narrator along with a viewer's drawing (shaded lines) of the Big Dipper overlaid with the correct outline (straight white lines). This application requires local interactivity by the user with a pointing device, such as a mouse, and a high performance procedural application environment. Hence the need for a DASE-PA like system.

8 Summary

This paper summarized the SMPTE DDE-1 specifications and compared DDE-1 to DASE. In addition, this paper compared DDE-1 applications ("enhancements") and DASE applications, and summarized the functionality specified within DDE-1 and DASE to enable content developers to present and manage ITV applications. This paper also gave examples of ITV applications.

Acknowledgements

The authors would like to thank Dr. Alan Mink and Mary Laamanen of NIST for their reviews of this paper.

9 References

[ATSC] Advanced Television Systems Committee, http://www.atsc.org

[ATSC-A90] ATSC Data Broadcast Standard, A/90, Advanced Television Systems Committee, 26 July 2000

[ATSC-ARM] ATSC Data Application Reference Model, ATSC ARM001R1, 17 Apr 2001

[ATSC-DATA] Chernock, et al, "Data Broadcasting: Understanding the ATSC Data Broadcast Standard", McGraw-Hill, 2001

[ATSC-IPM] DELIVERY OF IP MULTICAST SESSIONS OVER ATSC DATA BROADCAST, ATSC T3/S13-012

[ATVEF] Enhanced Content Specification, v1.1 r26, Advanced Television Enhancement Forum, http://www.atvef.com

[BW70] Business Week, October 4, 1999

[CSS1] Cascading Style Sheets, Level 1, W3C Recommendation

[CSS2] Cascading Style Sheets, Level 2, W3C Recommendation

DASE Documents:
ftp://atsc:atsc1080i@ftp.atsc.org/S17/T3/
ftp://atsc:atsc1080i@ftp.atsc.org/S17/

[DASE] DASE Part 1: Introduction, Architecture, and Common Facilities, ATSC T3/528, 9 Feb 2001

[DASE-API] DASE Part 4: Application Programming Interfaces, ATSC T3/531

[DASE-DA] DASE Part 2: Declarative Applications and Environment, ATSC T3/529, 9 Feb 2001

[DASE-CONF] DASE Part 8: Conformance Statement, ATSC T3/S17-xxx

[DASE-FONT] DASE Part 5: Portable Font Resource, ATSC T3/532, 9 Feb 2001

[DASE-PA] DASE Part 3: Procedural Applications and Environment, ATSC T3/530, 9 Feb 2001

[DASE-SEC] DASE Part 6: Security, ATSC T3/S17-xxx

DDE Documents: ftp://smpte.vwh.net/pub/d27/dde.htm

[DDE1] Declarative Data Essence, Content Level 1, SMPTE 363M

[DDE1-DOM0] Document Object Model Level 0 (DOM-0) and Related Object Environment, SMPT Draft q, 15 March 2001

[DDE1-IPM] Declarative Data Essence - IP Multicast Encapsulation, SMPTE 357M

[DDE1-lid] The Local Identifier (lid:) URI Scheme for Referencing Locally Stored Content Resources, SMPTE Draft g, 18 Apr 2001

[DDE1-NTSC] NTSC IP and Trigger Binding to VBI, DDE-1, SMPTE 361M

[DDE1-PAL] TRIGGERS and IP Binding to PAL/SECAM SYSTEM, SMPTE Draft a, March 2001

[DDE1-UHTTP] Unidirectional Hypertext Transport Protocol, SMPTE 364M

[DOM2-CORE] Document Object Model (DOM) Level 2 Core, W3C Recommendation

[DOM2-EVENTS] Document Object Model (DOM) Level 2 Events, W3C Recommendation

[DOM2-HTML] Document Object Model (DOM) Level 2 HTML, W3C Working Draft

[DOM2-STYLE] Document Object Model (DOM) Level 2 Style, W3C Recommendation

[DOM2-VIEWS] Document Object Model (DOM) Level 2 Views, W3C Recommendation

[ES2] ISO 16262, ECMAScript Language Specification, 2ed Edition

[ES3] ECMAScript Language Specification, 3rd Edition, ECMA

NIST DASE Development Environment:
http://www.itl.nist.gov/div895/cmr/dase/index.html

NIST DDE-1 Trigger Test Materials:

Abstract Test Suite: http://xw2k.sdct.itl.nist.gov/koo/trigger/

Sample Tape Documentation: http://xw2k.sdct.itl.nist.gov/koo/test_suite/

Sample Tape Test Cases: http://xw2k.sdct.itl.nist.gov/koo/test_suite/sample_tape.htm

[SMPTE] Society of Motion Picture and Television Engineers, http://www.smpte.org

[W3C] World Wide Web Consortium, http://www.w3.org

Mass Customization of DASE Broadcast and Increased Advertisement Capacity

Eddie Schwalb

Sharp Labs of America
eschwalb@sharplabs.com

With the advent of the Internet. the value of customizing content to the individual preference of millions of viewers became apparent, as the major Internet players, e.g., Yahoo!, derive a significant portion of their revenue from such mass-customization capabilities.

The advent of Digital TV has enabled advances in the area of user selectable content used in conjunction with traditional broadcast distribution. Unfortunately, existing DTV broadcasts still cannot be customized to the individual preferences of millions of TV watchers. In that respect, the multimedia broadcasting industry in general, and terrestrial broadcasting in particular, is lagging behind the Internet (aka .com) industry.

We present a method for customizing a single uniform broadcast to fit the preferences of individual viewers. The unexpected result is an arbitrary simultaneous increase in the advertisement effectiveness and time capacity, both of which are critical revenue drivers for broadcasters. Further, this is achieved without impacting, but rather protecting, the privacy of viewers, as it does not require communicating the preferences stored in the client back to the broadcasting server.

Mass Customization of DTV Broadcast

Eddie Schwalb

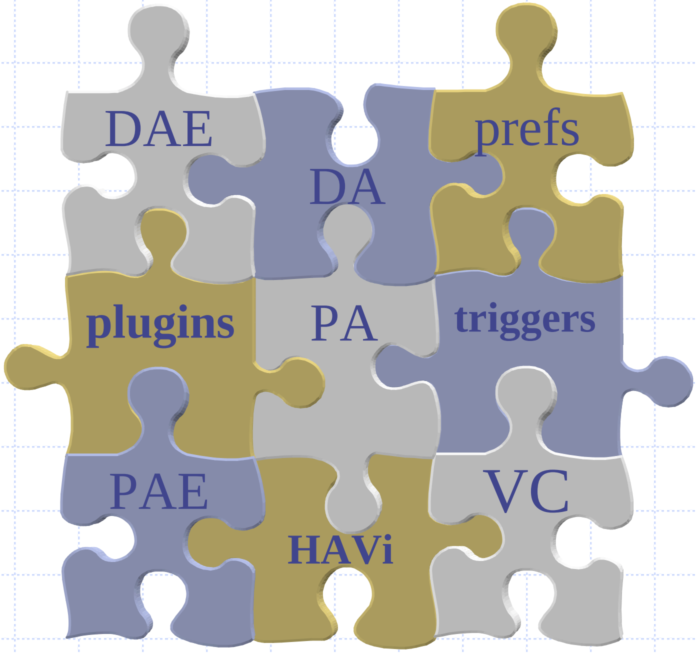
Sharp Labs of America

Introduction

- The Internet demonstrated the value of customizing content to the individual preference of millions of viewers
- What is required to achieve the same for Digital TV?
  - Can Direct Channel Change be used?
  - Can DASE be used?

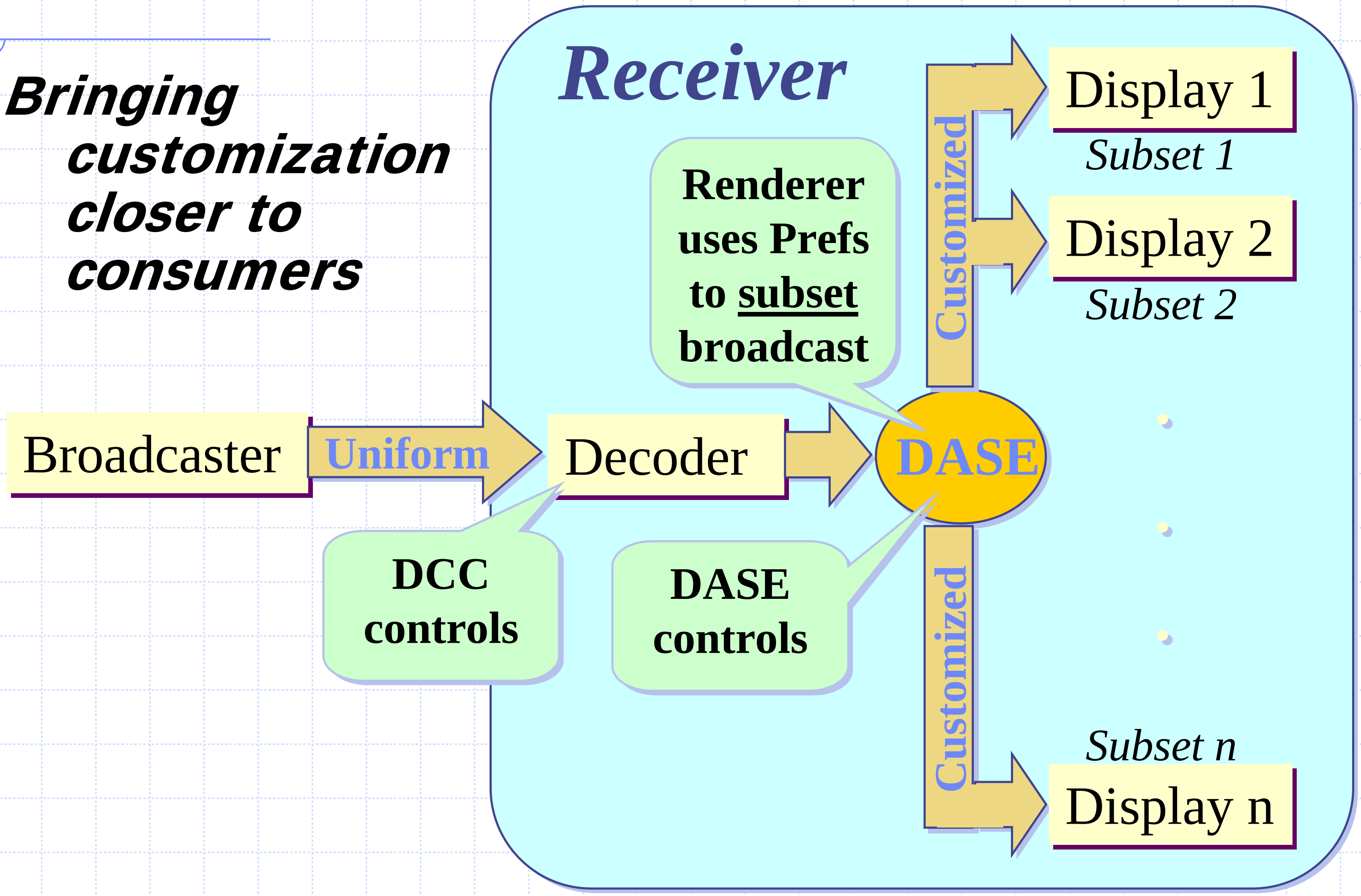
Outline

- Broadcast Customization
  - Transport Subsetting, Ad-Insertion
  - plugins, DCC, display switching, triggers
- Ramification:
  - increased capacity
  - happier consumers
- Roles
  - content authors
  - advertisers
  - broadcasters
  - receivers
- Summary



Transport Subsetting

Bringing customization closer to consumers



Ad-Insertion

- Static & Animated banner ads (Plugins)
- Display Switching (Native | Java)
- Synchronization (Triggers)

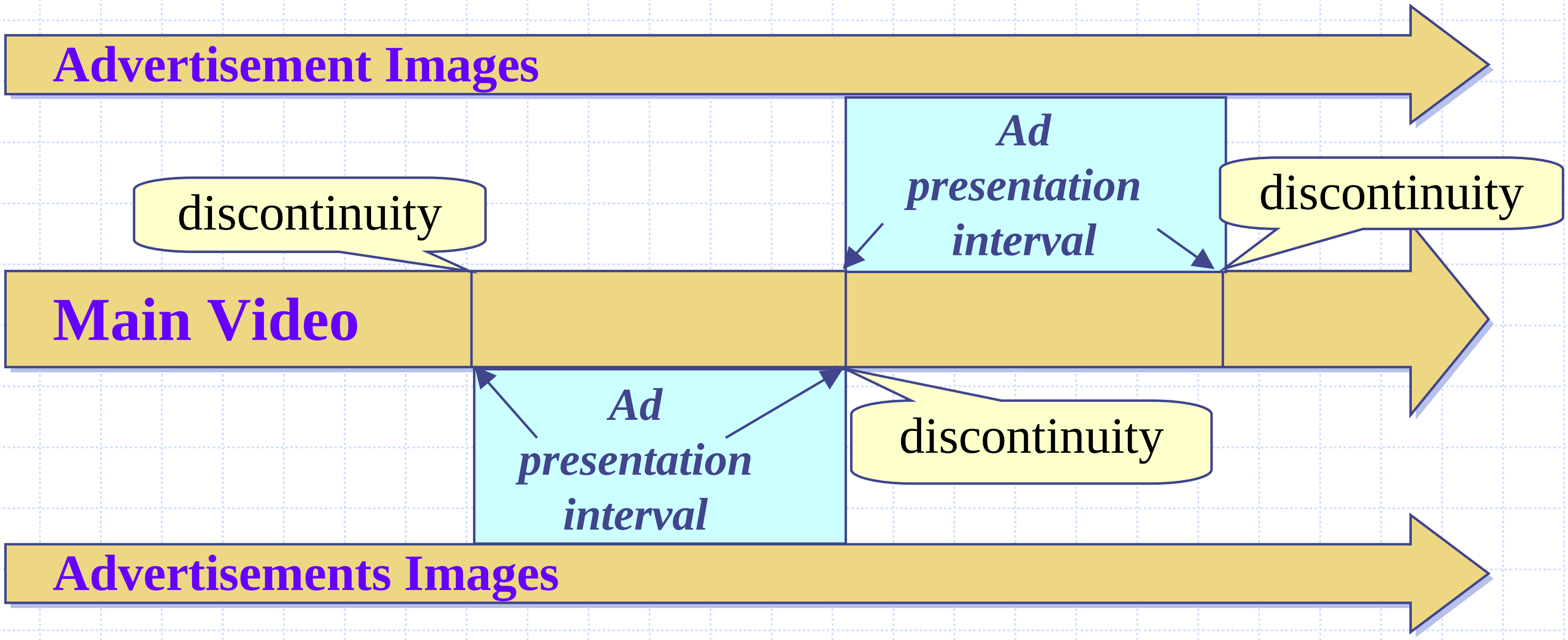
DA Ad-Selection using Plugins

```
<object classid="plugin.class“>
  <ARG.0=“banner-ad.png“ />
</object>
```

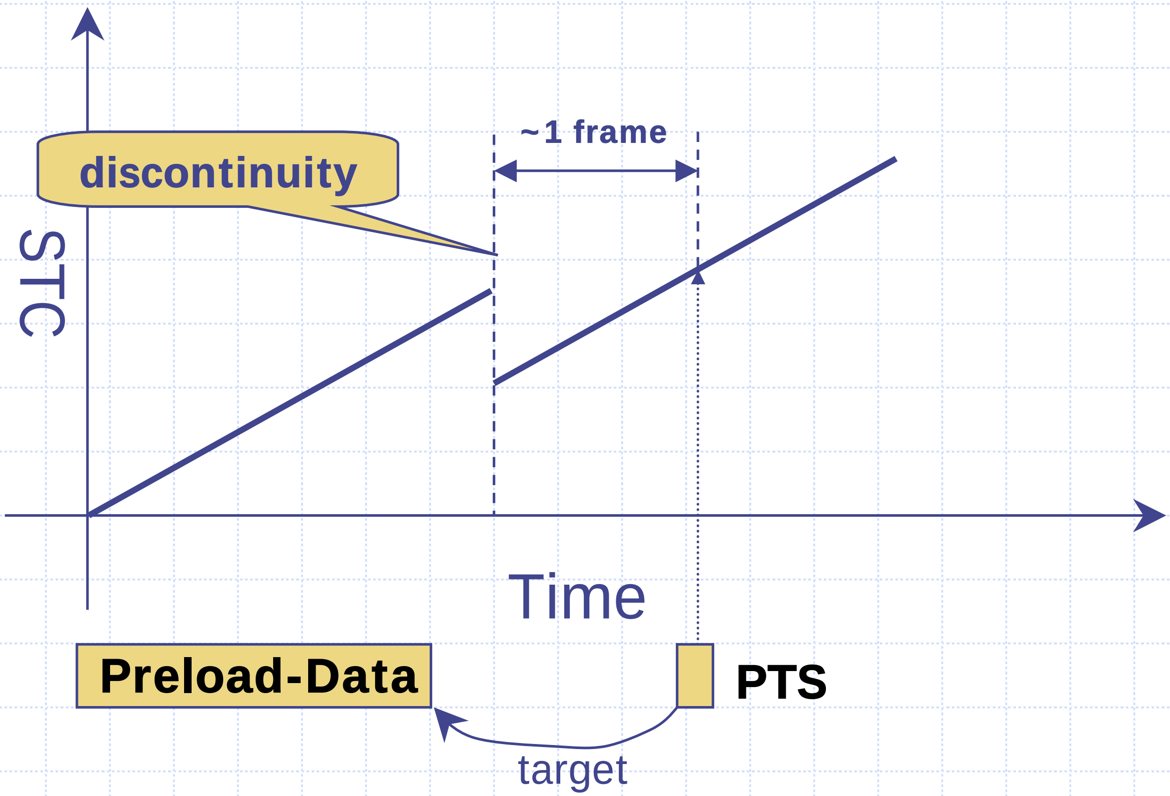
Plugin code queries user preferences ‘org.atsc.preferences‘ and customizes accordingly.

DASE Synced Ad Display

Banner-Ad insertion at discontinuity points

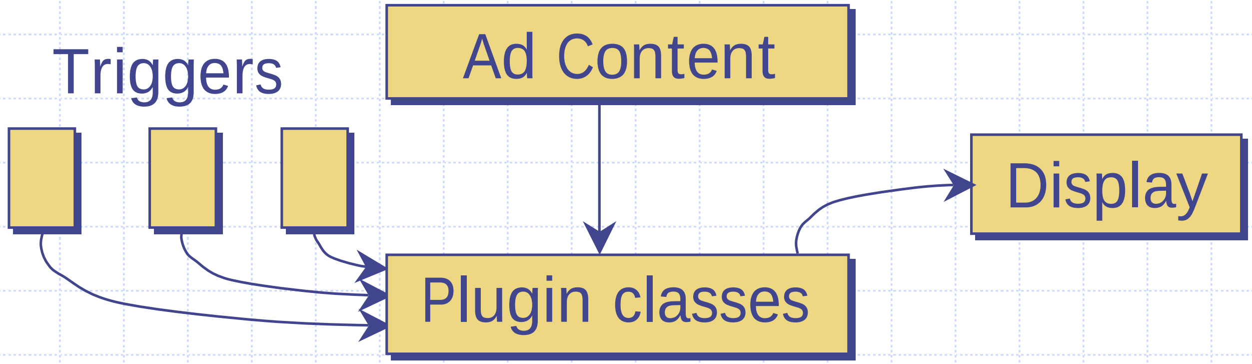


Trigger Model (S13 work-in-progress)



Plugins can process Triggers

```
<object classid="plugin.class“>
  <ARG.0=“banner-ad.png“ />
</object>
```

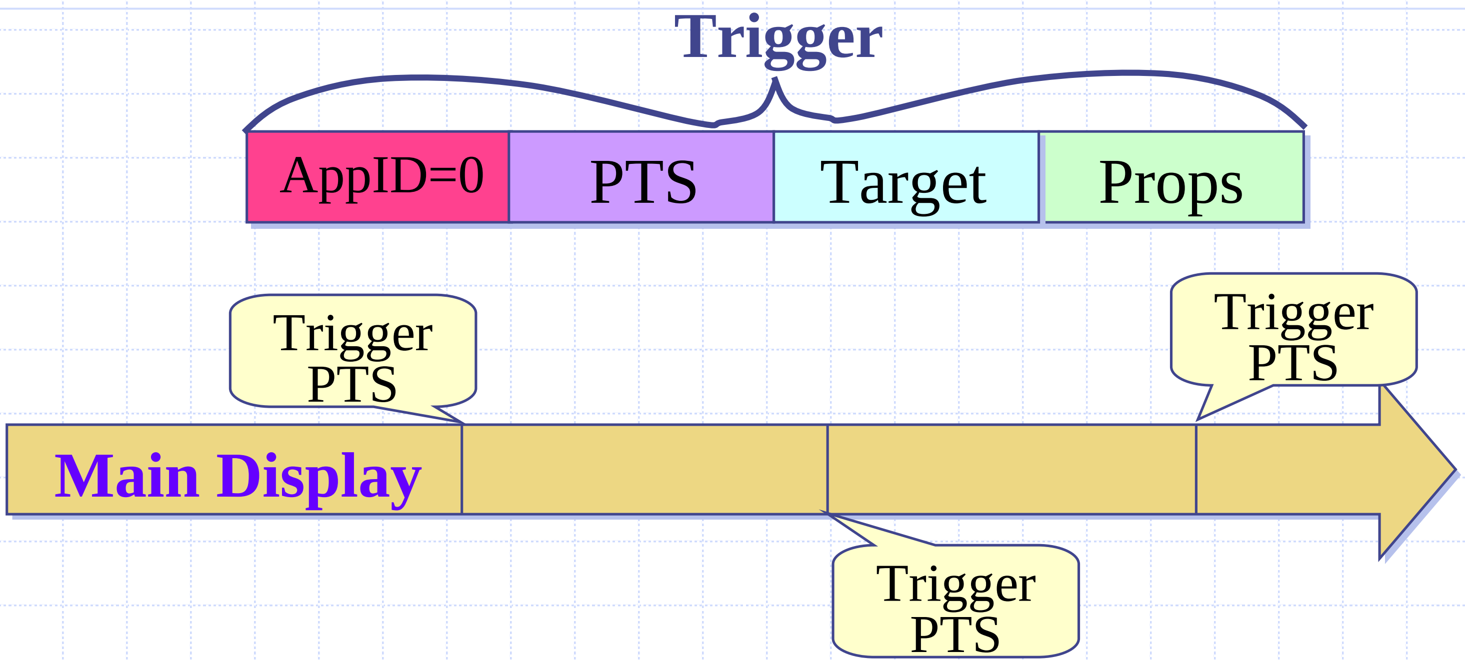


Plugin code can ‘listen’ for triggers via the TriggerAPI org.atsc.trigger.

DASE Triggers (work-in-progress)

Use Triggers to signal ad display:

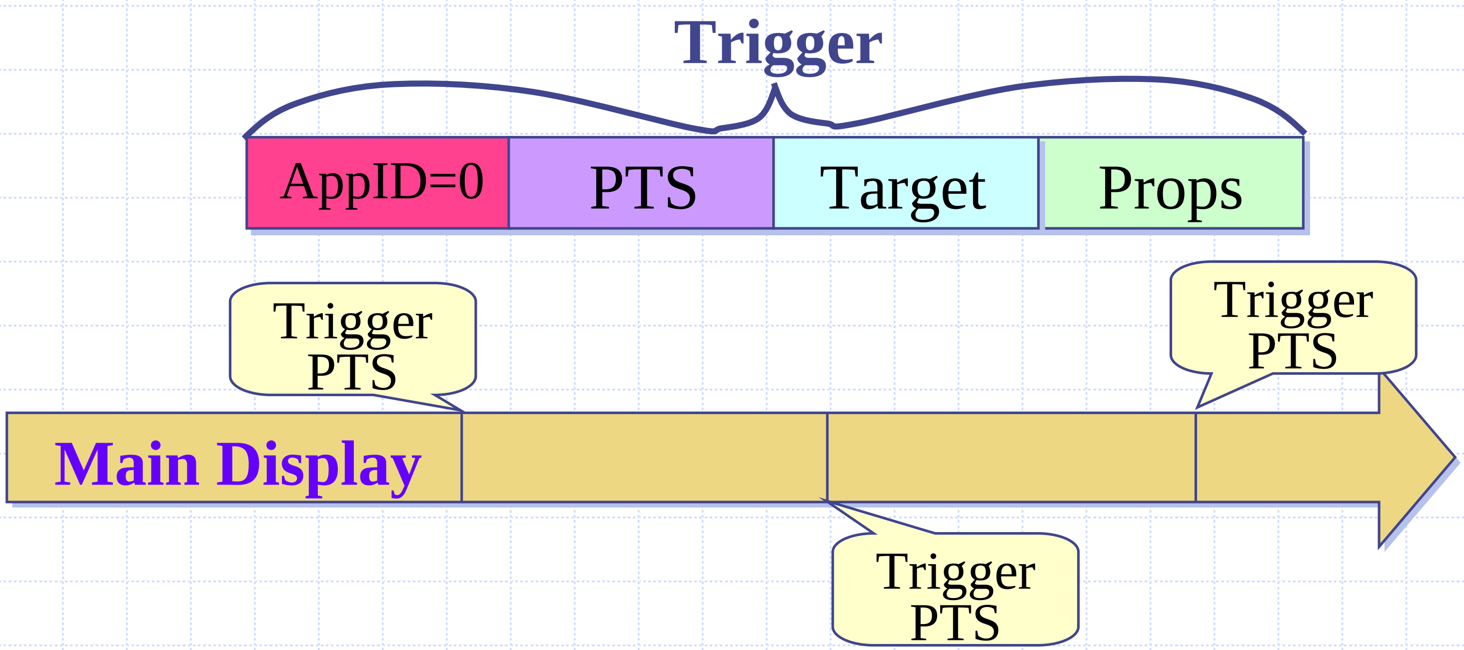
- AppID should be 0 to indicate ‘current’ service
- Presentation Time Stamp (PTS) set to display time
- Target contains reference to advertisement files
- Payload contains ‘key-words’,’properties’ information



DASE Triggers (work-in-progress)

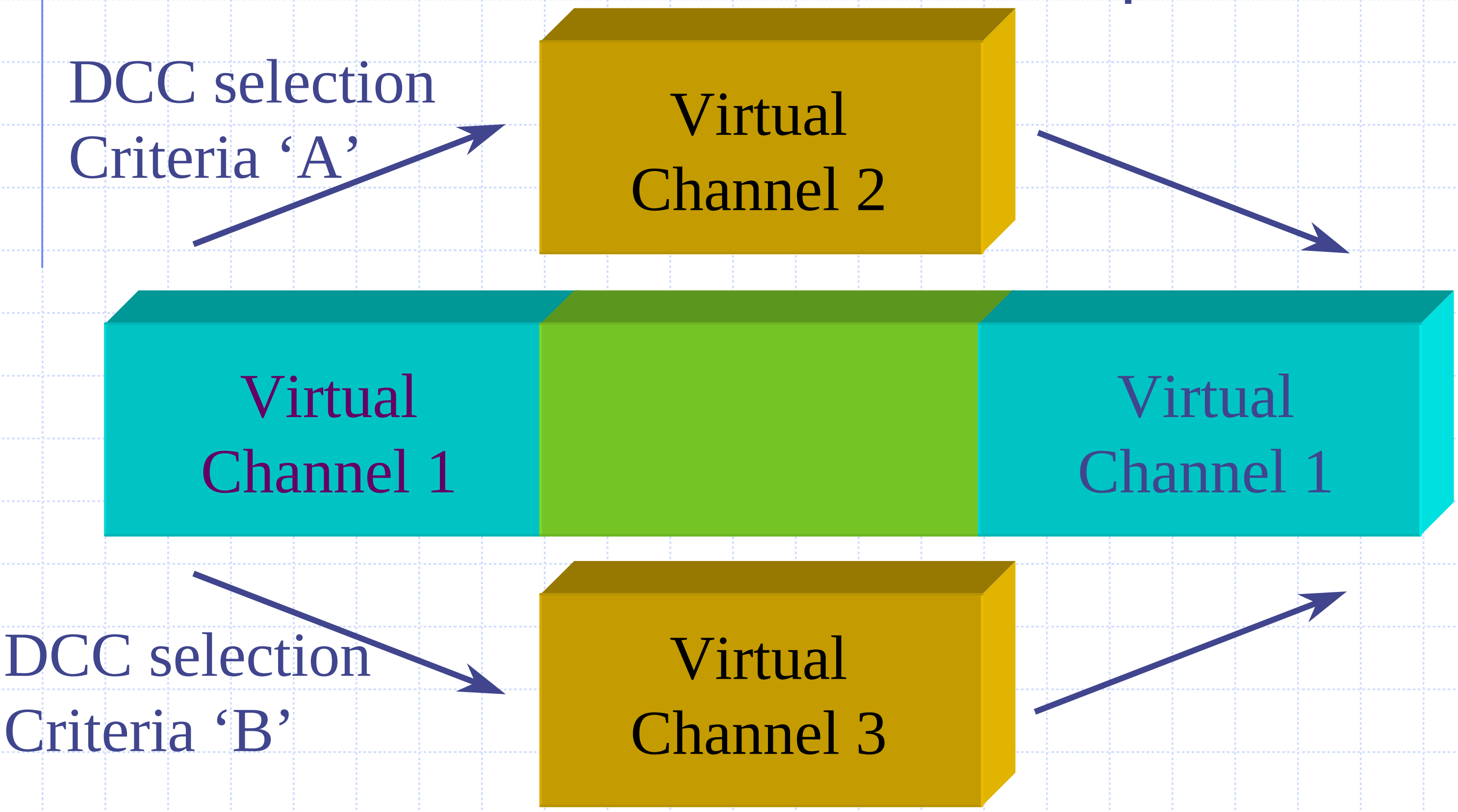
Use org.atsc.trigger.TriggerEvent:

- getTarget() to get advertisement files
- getProperties() to get ‘key-words’
- Use org.atsc.preferences to compare ‘key-words’



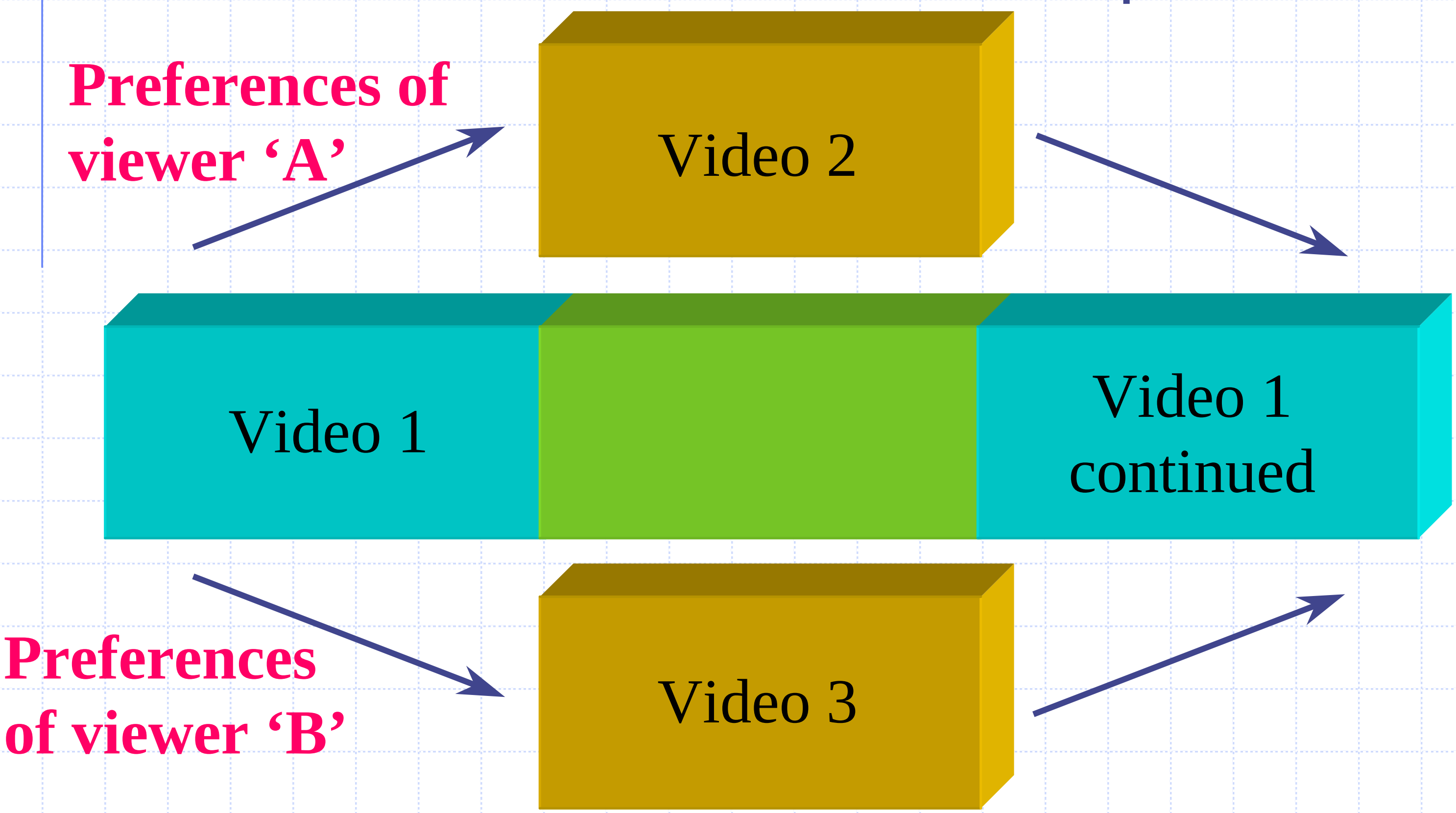
Direct Channel Change (why not)

DCC selection utilizes geographic criteria but does not utilize individual preferences



Display Switching
DASE improves on DCC

DASE enables applications to select video streams based on individual preferences

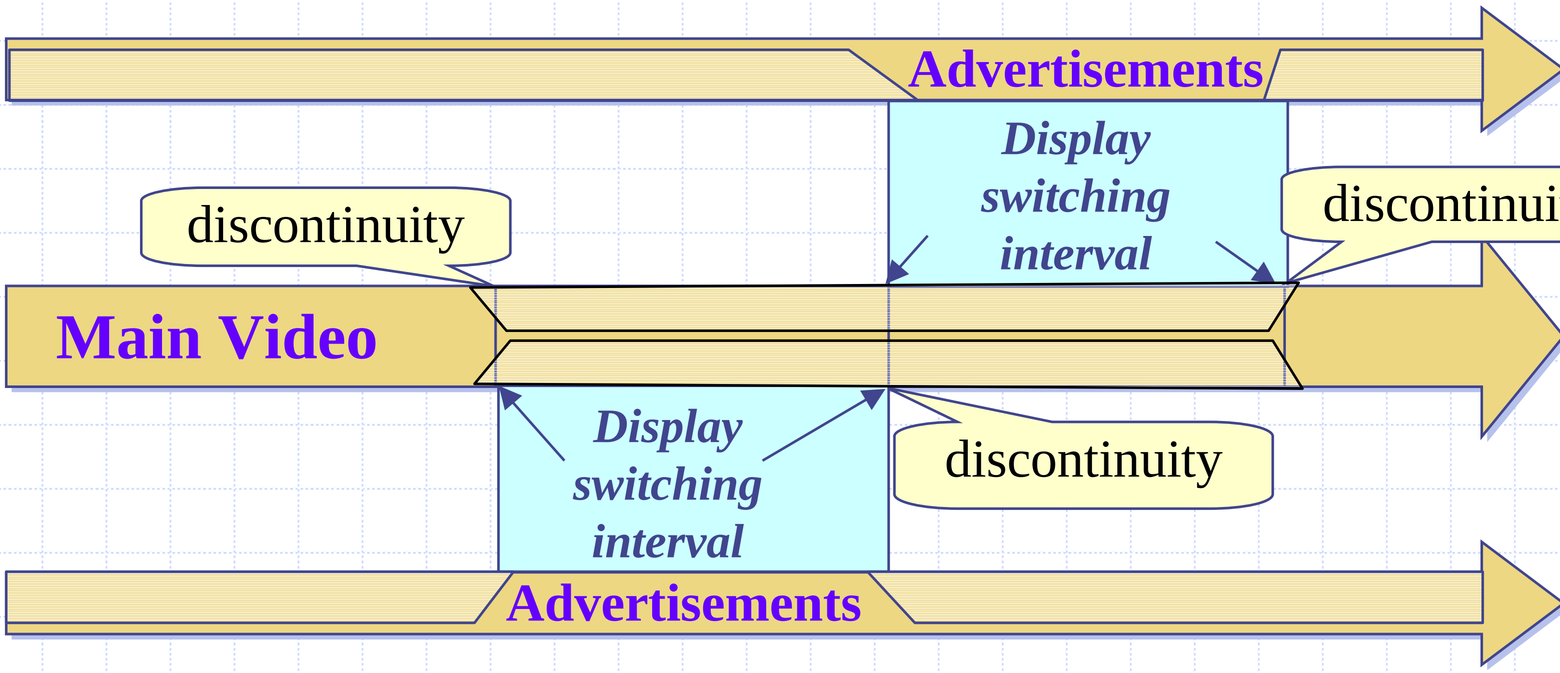


Tuning is performed via ServiceContextSelection

- Tuning could be native or Java
- JavaTV (or native emulation) enables decoupling of Services from their Display
- No ‘tuning’ API
- Service = Virtual Channel / Data Service
- ServiceContext = Display Area
- ServiceContextSelection
  - maps Service onto ServiceContext
  - = indirect ‘tuning’ via selection
- Selection Criteria: org.atsc.preferences

DASE Synced Display Switch

Plugin could 'switch' streams at points of abrupt change



Bandwidth, *bandwidth*, *bandwidth* …

So what?

Increased Capacity

- Assumptions:
  - Each commercial sequence has ***n*** slots.
  - There are ***k*** alternatives for each slot.
- Total number of possible sequences:

  n^k = exponential increase in possibilities.
- Each time slot:
  - Can be shared among ***k*** commercials
  - Each commercial is much better targeted
  - Value:
    - **Individual value may not decrease by factor *k* due to better targeting**
    - **Total value likely to increase having *k*-times more slots**
- Hypothesis: *Capacity increased in proportion with* ***k***

So what?

Happier Consumers

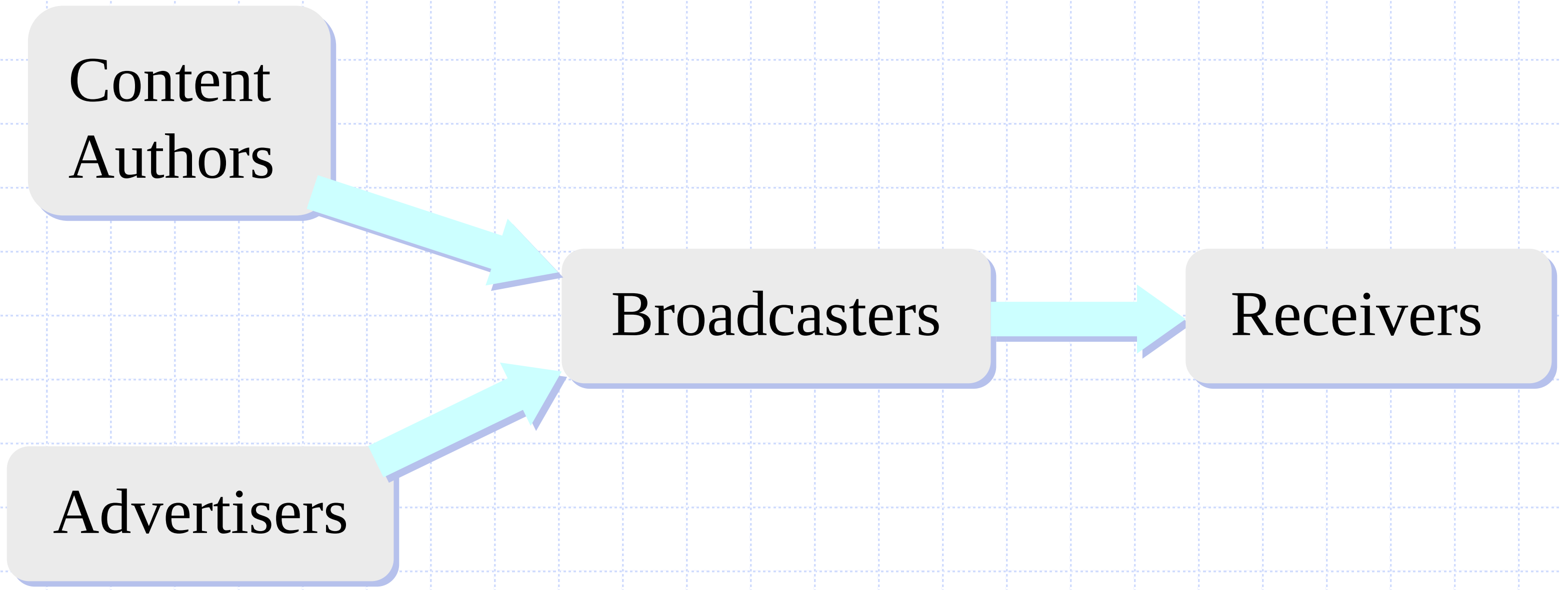
- Consumers could 'choose' their commercials (not eliminate them)
- Unlike the Internet – <u>private</u>

Better than Internet

- Customization is achieved without receivers sending preferences to emission stations

- Without a return channel, it is not possible to extract individual receiver preferences

Roles (simplified view)



Role of Content Authors

Should use triggers to mark commercial insertion points:

- PTS should point to insertion points
- Target & properties should be 'empty' as <u>hooks for broadcasters</u> to insert commercials
- Numerous hooks should probably be inserted per-insertion-point

Role of Advertisers

For each advertisement generate:

- key-words (trigger properties)
- PTS
- Aggregated Target

Role of Broadcasters

Should minimally process triggers:

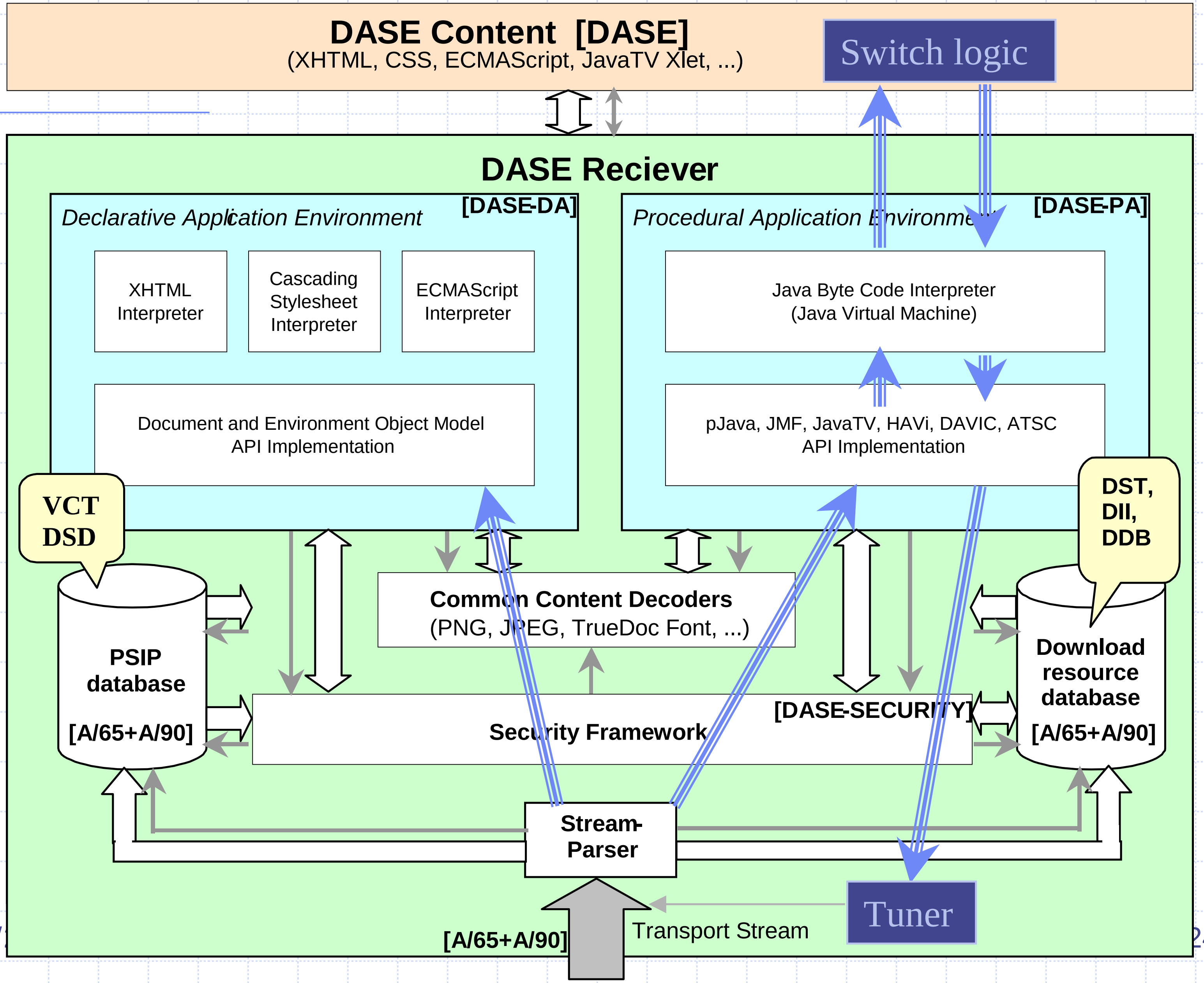
- populate trigger with references to commercial files
- populate trigger with key-words

Role of Receivers

Need to be capable of:

- collecting consumer's preferences
- matching key words of preferences with those of trigger's key words
  - DASE applications are launched before PTS
  - Listeners are invoked at the PTS
- switching display to the best matching video component
  - DASE applications perform ServiceContext selection

Architecture



Summary

- The promise of mass-customization could be transferred to broadcasting
- It is possible to simultaneously
  - increase advertisement capacity,
  - make consumers happier and
- DASE Triggers may be essential
- Requires some degree of coordination

Where to Get More Information

- S17: The DASE specifications
- S13: Data Broadcast specifications (A90)
- S18: Application Reference Model (ARM)

Comprehensive Public Key Infrastructures and the Realm of DASE Security

Edwin Heredia

Microsoft Corporation
1065 La Avenida,
SVC 3, Mountain View, CA 94043
eheredia@rnicrosoft.com

With the introduction of downloadable code in digital television -a media that can potentially reach millions of users simultaneously- comes the risk of spreading malicious code. The response is the deployment of a Public Key Infrastructure (PKI) that unfortunately cannot guarantee safety (it is extremely hard to determine if a piece of code is malicious or not) but at least, it establishes spread prevention, trust chains and, most importantly, legal bindings.

The objective of this presentation is twofold: first it will explain all (or most) of the components that are believed nowadays to be part of an overall comprehensive Public Key Infrastructure (PKI), and second we will offer a comparative analysis of the DASE security components against this comprehensive PKI. To offer a better perspective, we will include comparisons with other deployed PKI-based security architectures.

Full implementation of an end-to-end Public Key Infrastructure (PKI) requires agreed standards, certification authorities, structures among multiple certification authorities, methods to discover and to validate certification paths, certificate distributions, CRL distributions, management protocols for on-line and off-line interaction of PKI components, interoperable tools, provisions for certification, initialization, certification, revocation, recovery, and last but not least, supporting legislation. Implementation of all of these components would result in an overall comprehensive PKI that exists only conceptually but it is expected that practical implementations will approach this model in incremental steps.

Table 1 illustrates a selected list of comprehensive PKI elements. Because DASE-1 concentrates on a channel that starts at the emitter or broadcast station and ends at the decoder, the channel is considered relatively secure. However, if the server at the emitter becomes a gateway, or when a return channel is incorporated in the architecture, then an interoperable security model based on a PKI becomes necessary.

Table 1. Elements of a comprehensive PKI

Certification Authority	Certification Repository	Certification Revocation
Key Backup	Key Recovery	Automatic Key Update
Key History Management	Cross-Certification	Client software
Authentication	Integrity	Confidentiality
Secure Time Stamping	Notarization	Non-repudiation
Secure Data Archive	Privilege/Policy Creation	Privilege/Policy Verification

A Comprehensive Public-Key Infrastructure and the realm or DASE security

Edwin Heredia

Microsoft Corporation
Silicon Valley Campus, Bldg 3
1065 La Avenida,
Mountain View, CA 94043

PKI functions

Basic PKI objectives

- ❑ Endpoint Authentication
- ❑ Message Integrity
- ❑ Confidentiality

It is extremely difficult to guarantee the safety of downloadable code. At most we can provide security tools to minimize the problem and establish a chain of trust for legal bindings.

Additionally, a PKI provides

- ❑ Spread prevention of unsafe code
- ❑ Trust models that depend on registered authorities
- ❑ Legal bindings through digital signatures

These functions are implemented using public-key cryptography, thus the name of Public-Key Infrastructure (PKI).

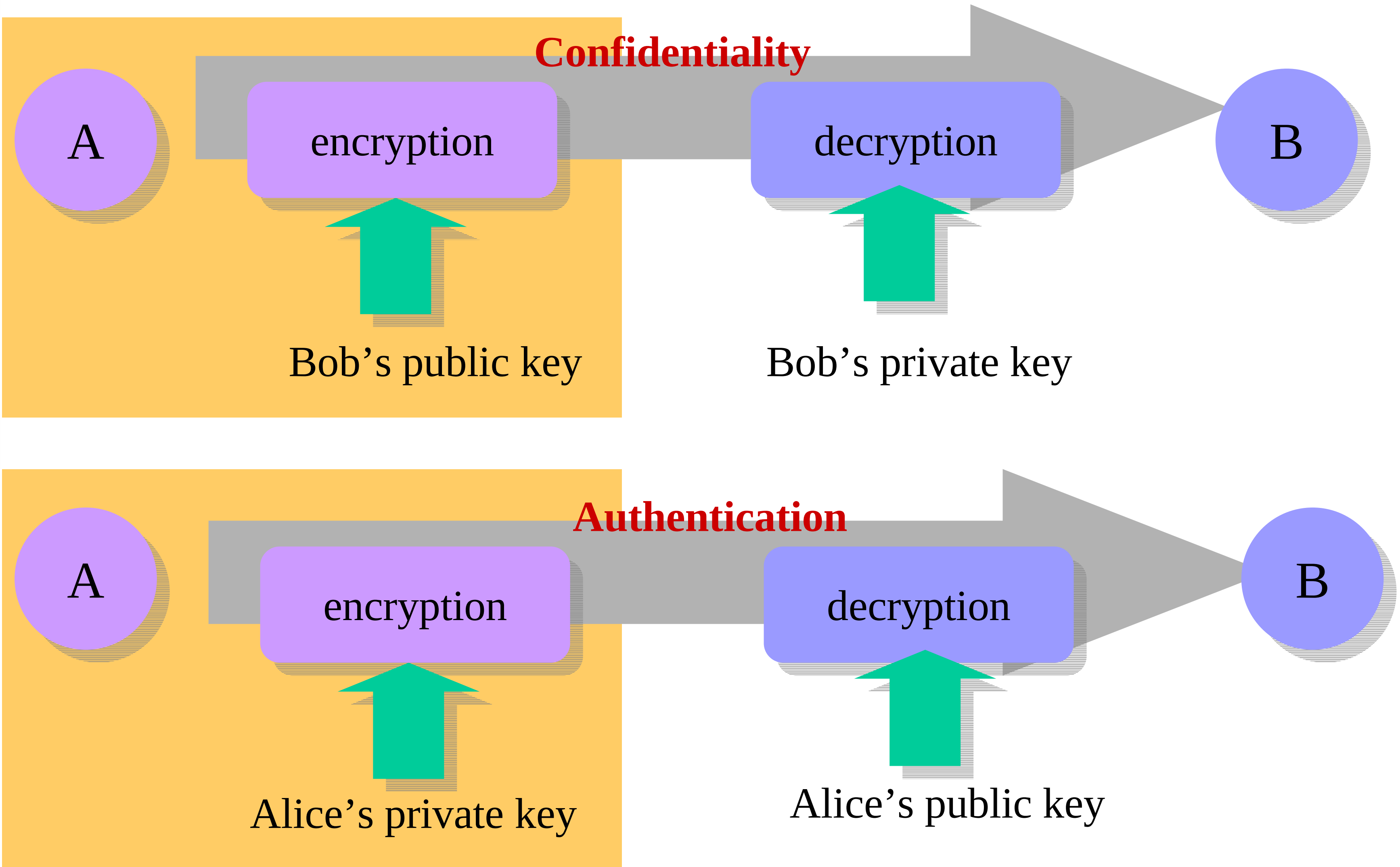
Threat Models

- What resources we expect an attacker to have available?
- Every security system is vulnerable to one threat or another
- What attacks are we going to worry about?
- What attacks we are NOT going to worry about?
- Protecting against attacks where one of the end systems is under control of the attacker is extraordinary difficult
- An active attack is when the attacker writes/modifies content. A passive attack merely involves reading data from the network.
- Usually we do not worry about denial-of-service attacks, not because they are not important, but because they are very difficult to protect.
- Security should not become more expensive than what it is worth

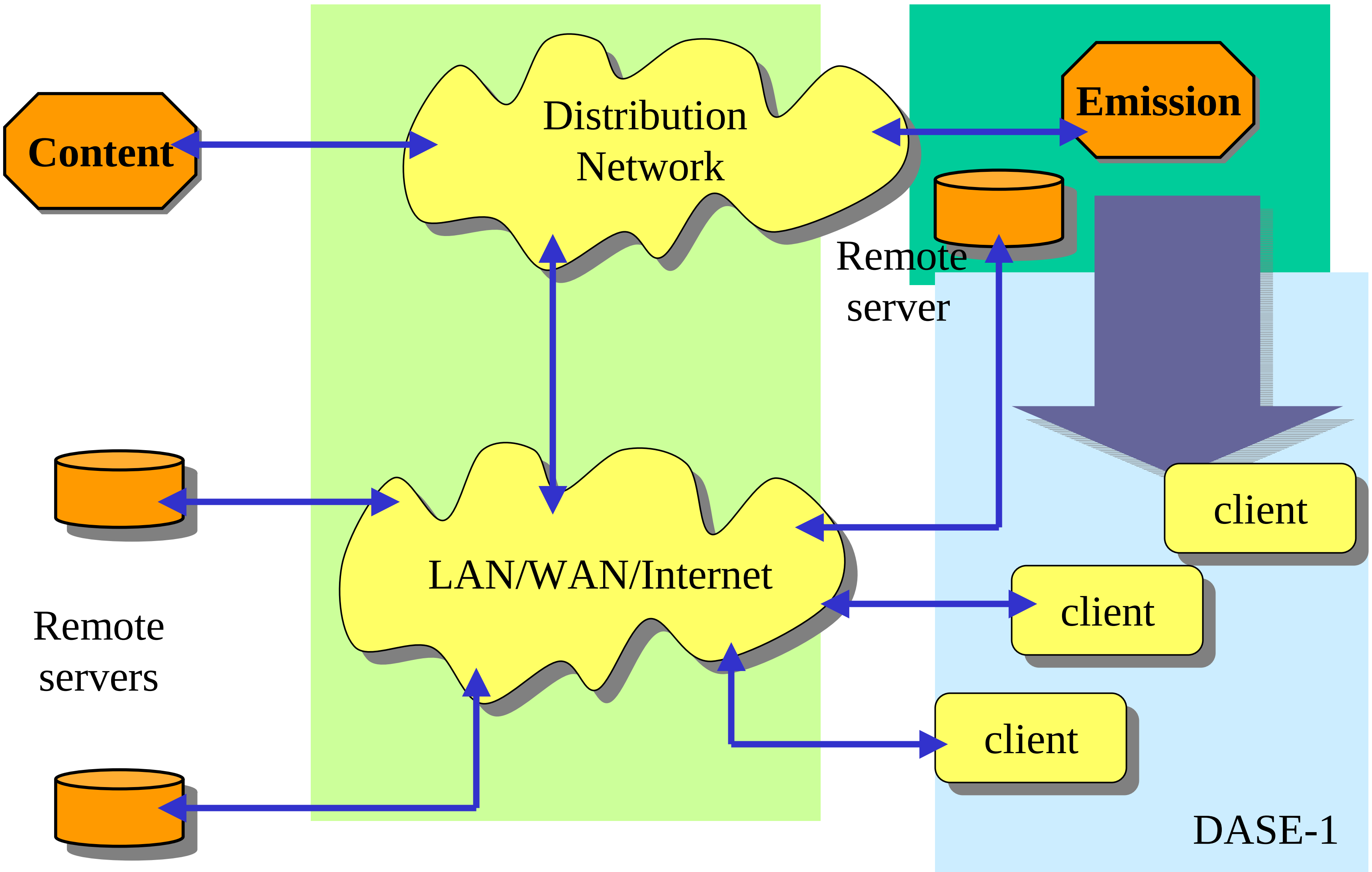
Cryptographic Algorithms

Symmetric encryption	RC2, RC4, RC5, DES, 3DES, IDEA, AES
Digest algorithms	MD-5, SHA-1
Key establishment	RSA, Diffie-Hellman
Digital signature	RSA, DSS (based on RSA, DSA or EC-DSA)
Message authentication	DES-MAC, HMAC

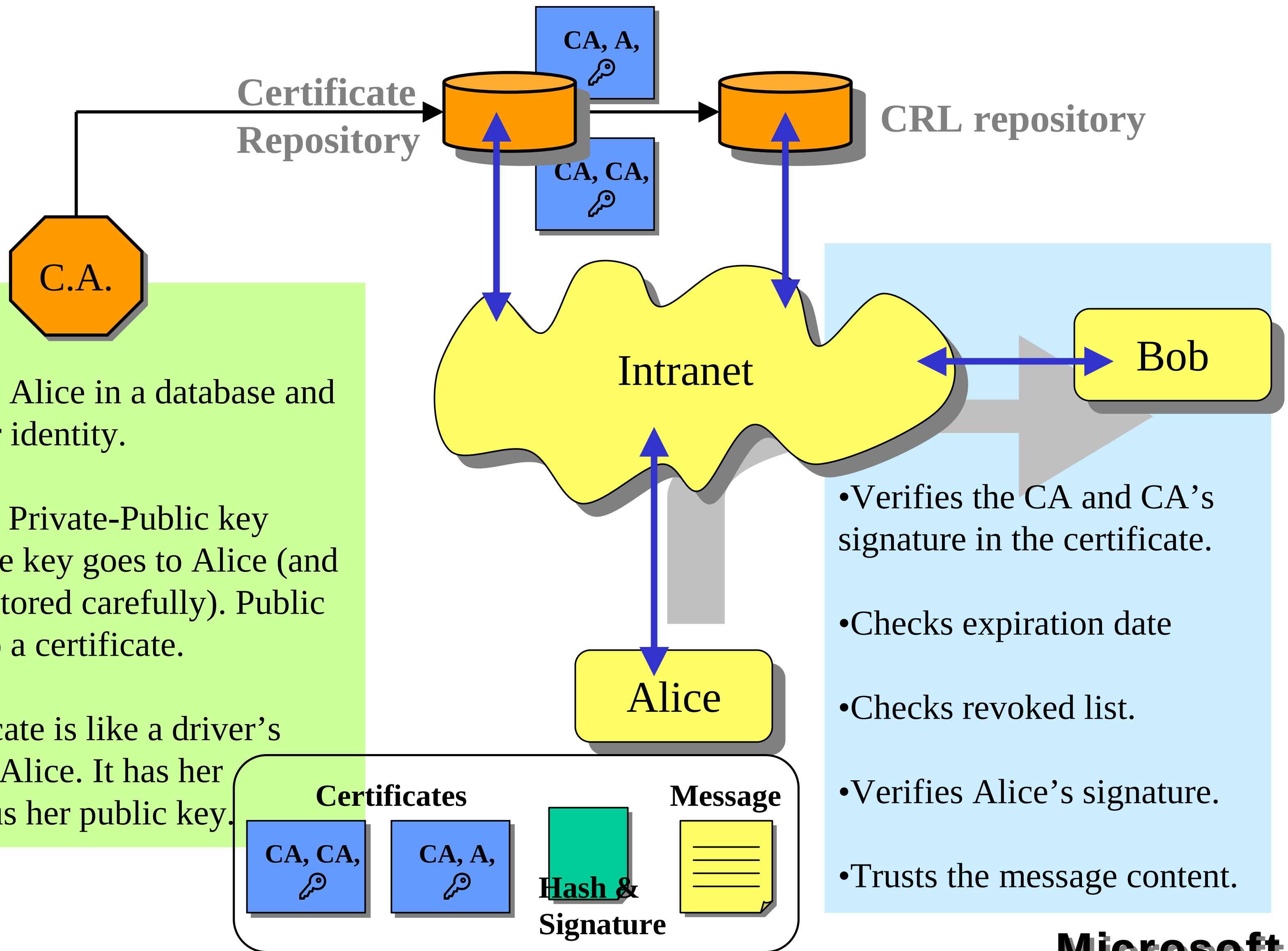
PKI cryptography



DTV Infrastructure



Public Key Infrastructure



✓Registers Alice in a database and verifies her identity.

✓Creates a Private-Public key pair. Private key goes to Alice (and should be stored carefully). Public key goes to a certificate.

✓A certificate is like a driver's license for Alice. It has her identity plus her public key.

- Verifies the CA and CA's signature in the certificate.
- Checks expiration date
- Checks revoked list.
- Verifies Alice's signature.
- Trusts the message content.

Microsoft

Roles and Responsibilities

CA Responsibilities

- Use a trustworthy system and provide a secure environment
- Disclose practices and procedures
- Properly identify certificate applicants
- Publish issued certificates
- Revoke certificates
- Make warranties to the certificate applicant upon issuance of the certificate
- Make warranties to persons using the certificate to verify digitally signed messages

Subscriber Responsibilities

- Make truthful representations in applying for a certificate
- Review and accept certificates before using them
- Make certain representations upon acceptance of the certificate
- Control and keep confidential the private key
- Report key compromise as soon as it happens

Elements of a comprehensive PKI - 1

Element	Internet	SSL client	Intranet	DASE
Certificate authority	yes	yes	yes	2
Certificate repository	no	no	yes	2
Certificate revocation	no	yes	yes	2
Key backup	no	yes	yes	no
Key recovery	no	no	yes	no
Automatic key update	no	no	yes	no

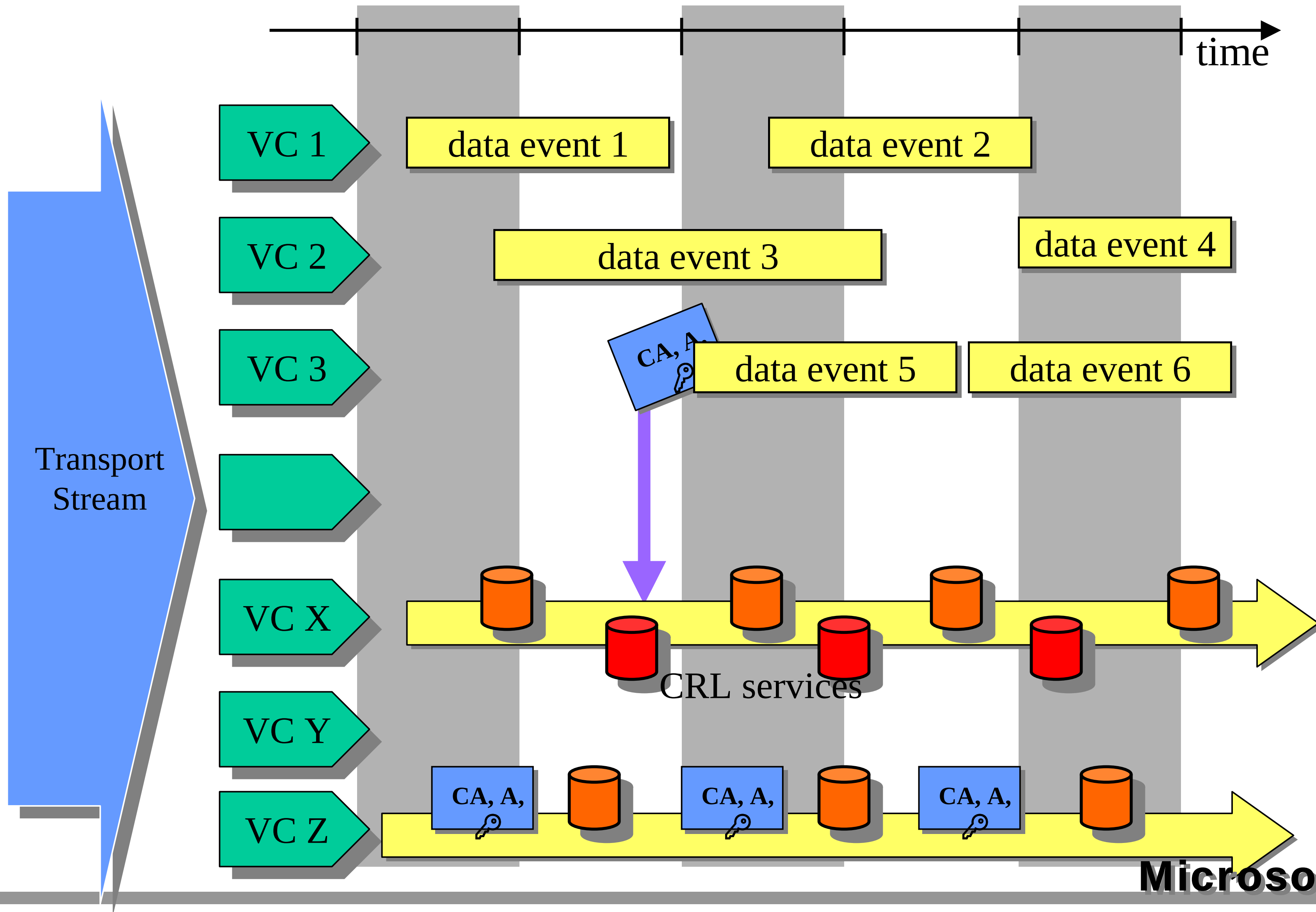
Elements of a comprehensive PKI - 2

Element	Internet	SSL client	Intranet	DASE
Key history management	no	no	yes	no
Cross-certification	no	no	no	2
Client software	no	no	yes	2
Authentication	yes	yes	yes	2
Integrity	yes	yes	yes	2
Confidentiality	yes	yes	yes	2

Elements of a comprehensive PKI - 3

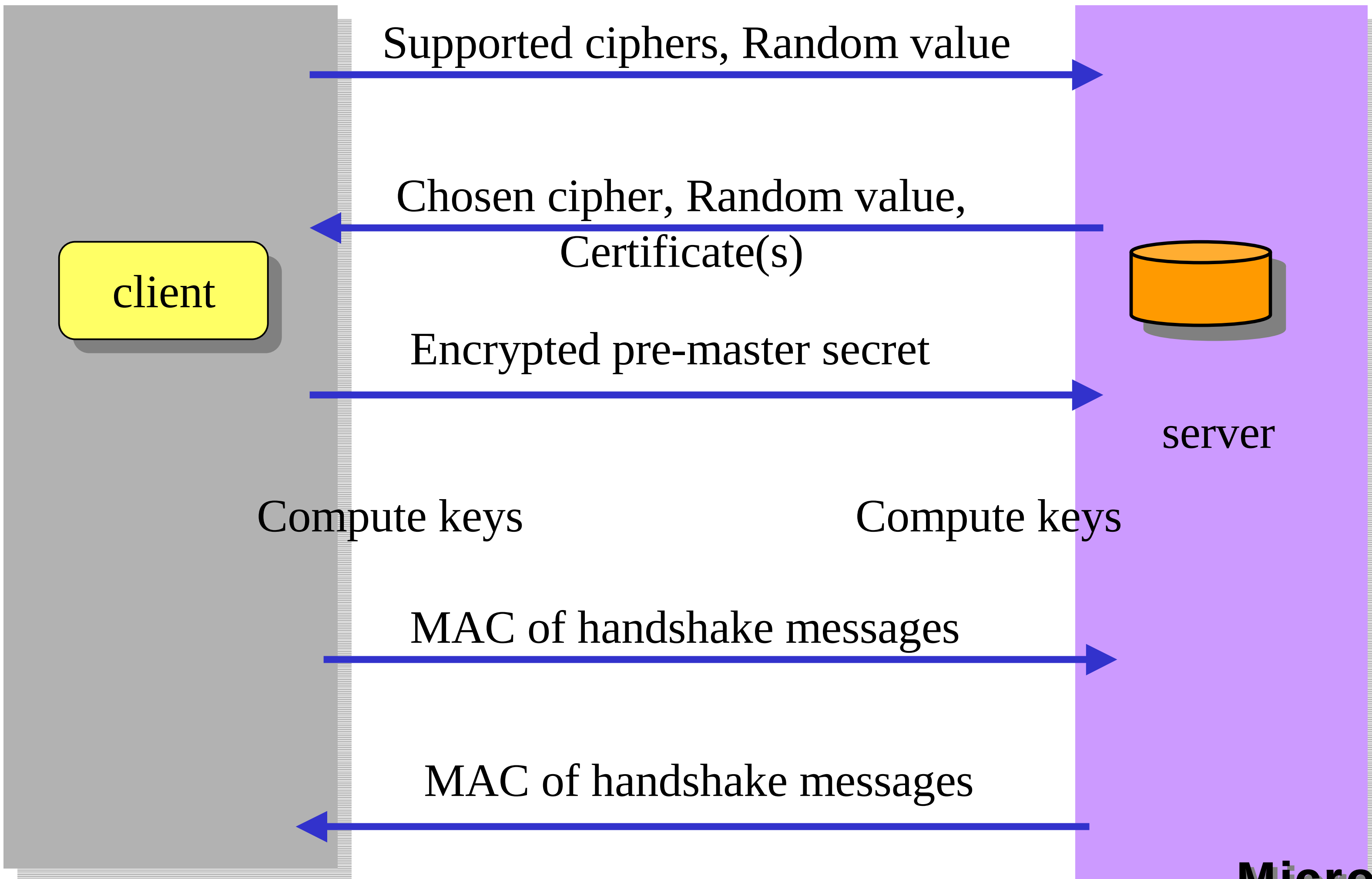
Element	Internet	SSL client	Intranet	DASE
Secure time stamping	no	no	no	no
Notarization	no	no	no	no
Non-repudiation	partial	partial	partial	partial
Data archives	no	no	no	no
Privilege/policy creation	no	yes	no	2
Privilege/policy verification	no	yes	no	no

Broadcast Transport Model



Microsoft

SSL/TLS as a model for Return Channel security



Certificates

- Based on X.509 v3 standard
- Serves as the legal binding between a subject and its public key
- Issued by a legal certificate authority that must verify and register the identity of the subject.
- Certificates include a lifetime period.
- The issuer needs to digitally sign the certificate.
- If the issuer is not a root CA, it needs to provide extra certificates to link with the established root CA.
- Other important fields include the subject's alternative name, the usage of this key, CRL distribution points, certificate policies, and others.
- May be delivered with the application or stored in a repository

Certificate Revocation Lists

- Used to revoke a certificate before its expiration.
- A CRL is a list of certificate serial numbers.
- Typically updated every few weeks (this period depends on how sensitive the information is).
- Under strict control and ownership of the C.A. - The lists are signed by the C.A. - Repositories are maintained (and secured) by the C.A.
- Syntax and Semantics defined in X.509 v.2.
  - CRL common fields (version, signature, issuer, this/next update).
  - Certificate entries and entry extensions.
  - CRL extensions.
- "Next Update" field in CRL may be used by hosts (or STBs) to cache the latest release at the proper time.
- A multicast/broadcast channel is a perfect distribution method for CRLs. However, they are normally accessed on demand. They may be accessed in real time.

Standards that use PKI

PKIX	A working group for IETF that designs interoperable PKIs for the Internet
PKCS	De facto standards for public-key message exchange.
TLS/SSL	Provides a secure and authenticated channel between hosts and the Internet above the transport layer.
IPsec	Defines transparent encryption for network traffic.
Kerberos V5	Provides a symmetric key-based framework in large networks.
S/MIME	Provides a standard for secure e-mail
DVB	Provides a multimedia platform with security as a method to authenticate and check integrity for apps

Thanks !

Where do you want to go today?®

http://www.microsoft.com/TV

Aggregating DASE Applications

Eddie Schwalb

Sharp Labs of America
eschwalb@sharplabs.com

When authoring DASE content, applications need to be packaged within data services. However, such packaging requires tight coordination and collaboration between content producers and content aggregators. Key issues include the distribution of applications among data services, URI mappings, integration of triggers and startup Gill integration. We explore some issues that need to be addressed in order to **enable content-producers to pre-package broadcast-ready** (i.e., shrink-wrap) **applications.**

We present a framework of requirements with which compliance enables the decoupling of content producers from aggregators, and allows aggregators to transfer data to emission stations without tight coordination. We suggest that the packaging should be in a format that could be automatically analyzed and placed on ATSC-compliant transmission systems. Methods are presented on which content-producers can rely when authoring applications, content-aggregators can rely when compiling data services, and content emitters can rely on when emitting content to be viewed by millions of viewers. The requirements are presented in the form of constraints on the structure and content applications.

The distribution of applications within data services involves tradeoffs between bandwidth and accessibility. Aggregating all applications into a single data services renders them all available without the need for broadcasters to maintain consistency of multiple related data services and without the need for receivers to switch data services (requires special permission) when switching applications. However, the use of carousels implies that as their size becomes large enough, the response time of receivers (downloading these carousels) becomes prohibitively slow.

Application resources are associated with URIs, which need to be embedded within the transport stream in appropriate locations, e.g., either DST or NRT. An automated procedure is needed, that analyzes applications and generates the transport binding for the URIs used in all inter-related applications. While this problem is manageable for Declarative Applications (DA), it much more difficult for Procedural Applications (PA). To solve this problem, additional mapping information needs to be shipped with the PA to the aggregator.

Integration of triggers is a complex issue. The transport layer defines lightweight triggers that, instead of carrying much data or code, carry pointers to the relevant data and code. In contrast, DASE requires the use of heavyweight triggers that carry both data and code. This difference implies the need for a packaging phase, in which heavyweight DASE triggers, generated by content-authors, are translated into lightweight transportable triggers.

The data service startup Gill application could serve as a dispatching application which is the target for Gill events and triggers. Because DASE-l allows a single application to execute at any given time, an inevitable challenge is managing application switching. While solving this problem, we uncovered a number of constraints and structures that need to be imposed on content producing tools, to be used by content-producers and aggregators alike.

Aggregating DASE Applications

Eddie Schwalb

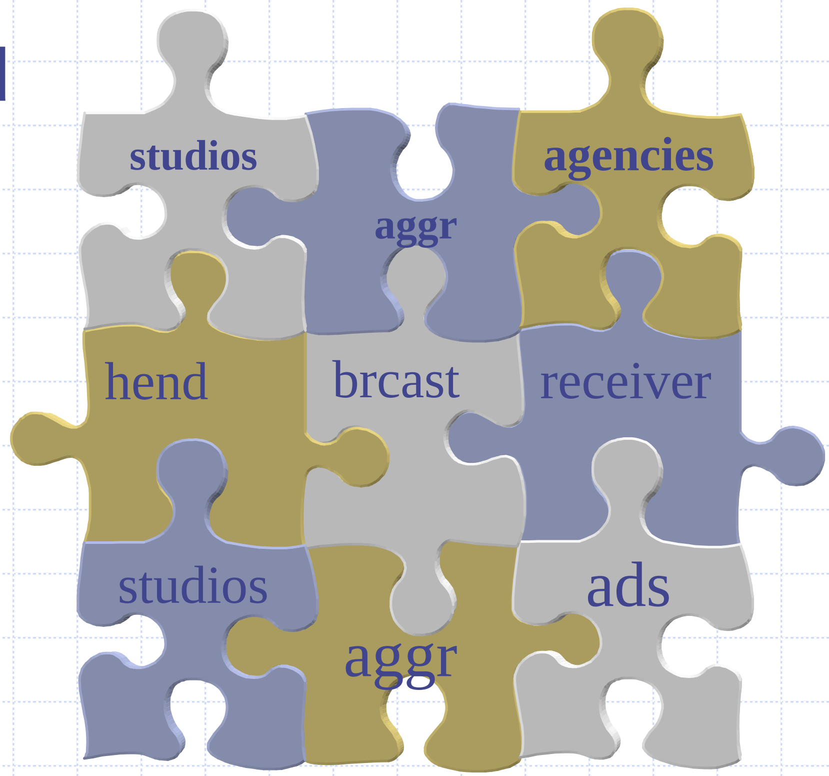
Sharp Labs of America

Introduction

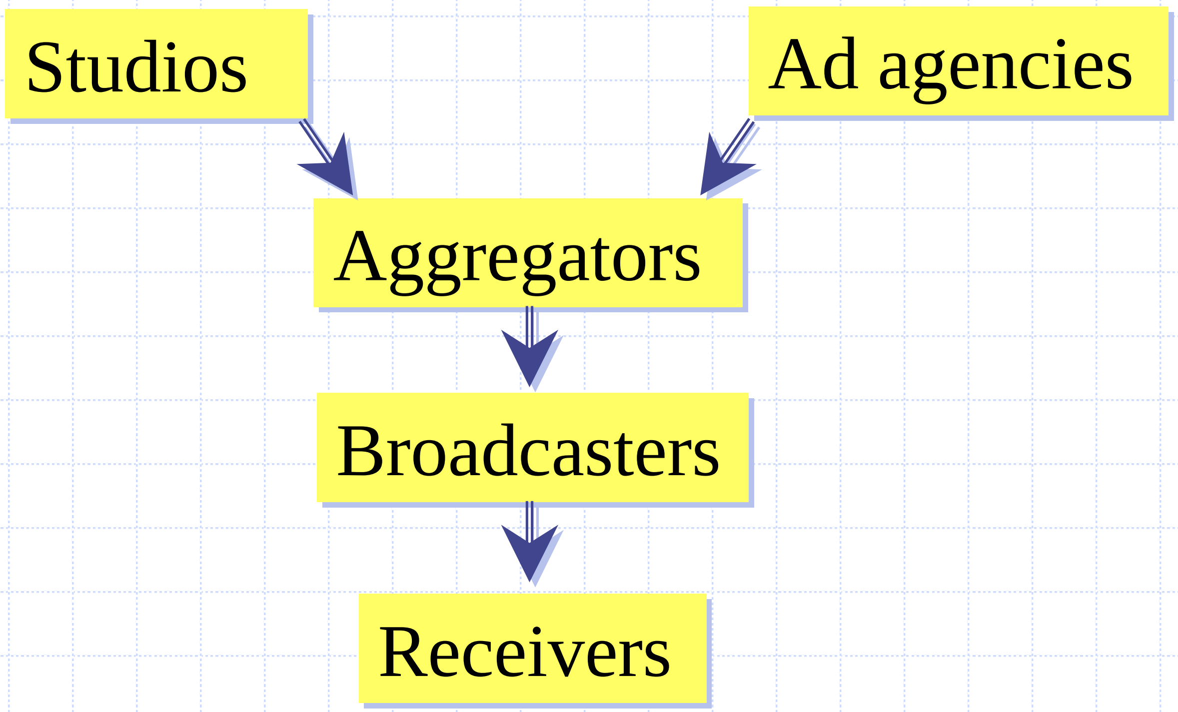
- DASE Application must go through a long process until they reach a consumer
- Various aggregation tasks are presented
  - The aggregation model
  - URI mapping tasks
  - Meta-data usage
  - Dispatching application
  - Bandwidth considerations

Overview

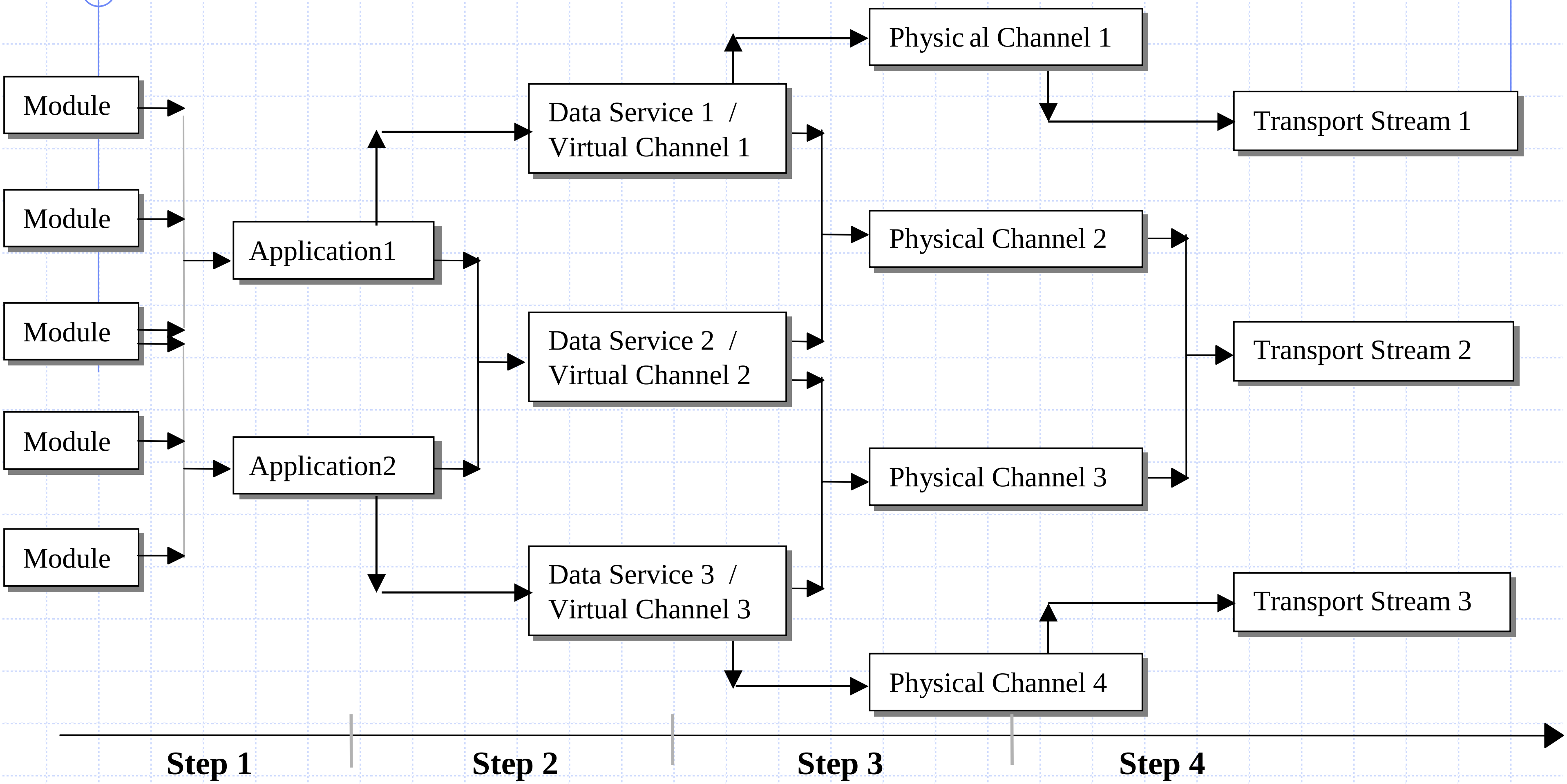
- The ATSC standard defines how applications are structured and placed in a transport stream.
- Impacted are authors, aggregators, broadcasters and receiver manufacturers.



Context ‘food-chain’



The Aggregation Model



The aggregation Problem

Aggregate:

- modules (code+data) into applications
- applications into data services (virtual channels)
- data services into major channels & transports

... each of these steps requires some manipulation of content

ATSC Impact

- Studios & Add Agencies:
  - Defines content specifications
    - XDML is used to author DTV-pages
    - JavaTV and org.atsc API used for active content
- Aggregators:
  - defines an aggregation model
    - URI name spaces
    - Relationships between modules and transports
- Broadcasters:
  - Defines the transport specifications
    - uses A90 to define transport-layer
    - uses ARM for an application reference model
- Receiver manufacturers:
  - Defines receiver behavior
    - receiver architecture
    - behavior of execution environments

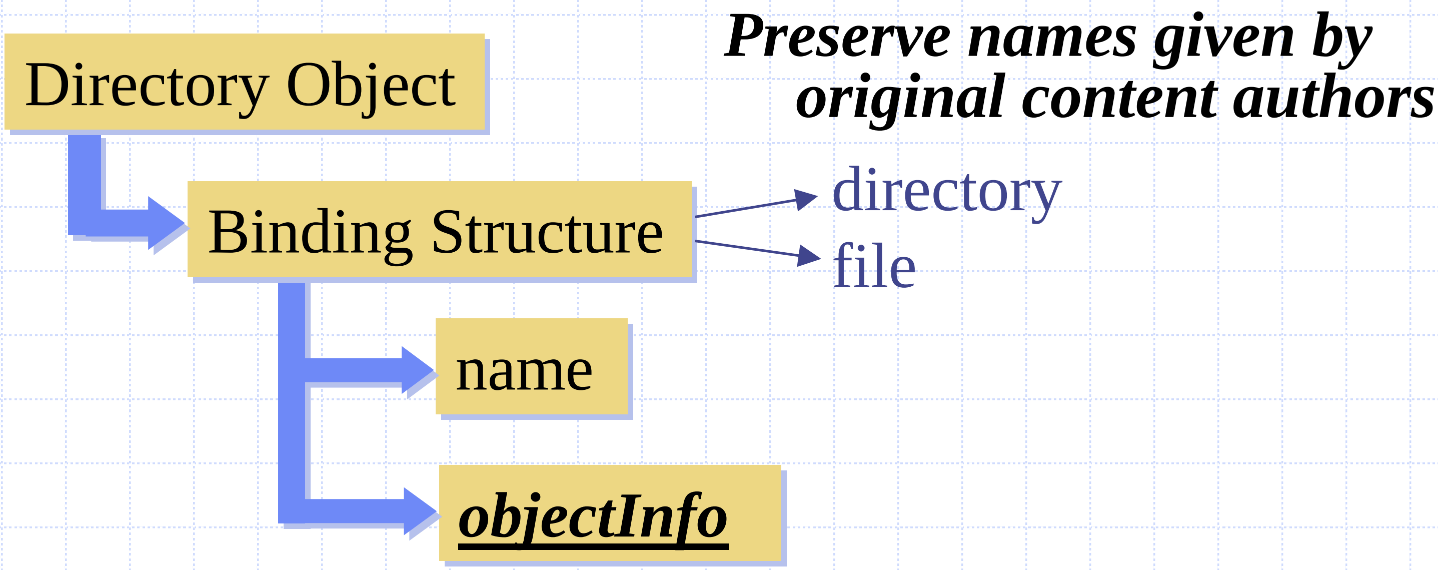
URI usage (studios)

- Modules are assigned a lid URI at authoring time
- lids of modules are aggregated in each step to form a complete application and data service/virtual channel hierarchy
- lids should be unique to avoid collisions
- lids should **not** include transport-layer configuration:
  lid://transport/channel/service/application/module/file.type

URI extraction
(studios, ad-agencies & aggregators)

- Applications comprise of
  - Resources (data-essence)
  - URIs (meta-data)
- Aggregators could extract URIs from static references made by DA
- It is not possible to extract URIs generated by dynamic ECMA Script or Java code.
- Both studios and ad-agencies need to attach to their content application an exhaustive list of URIs that are to be placed in a program by the aggregators.
- All that is really needed is management of base-URIs.

Aggregating Files



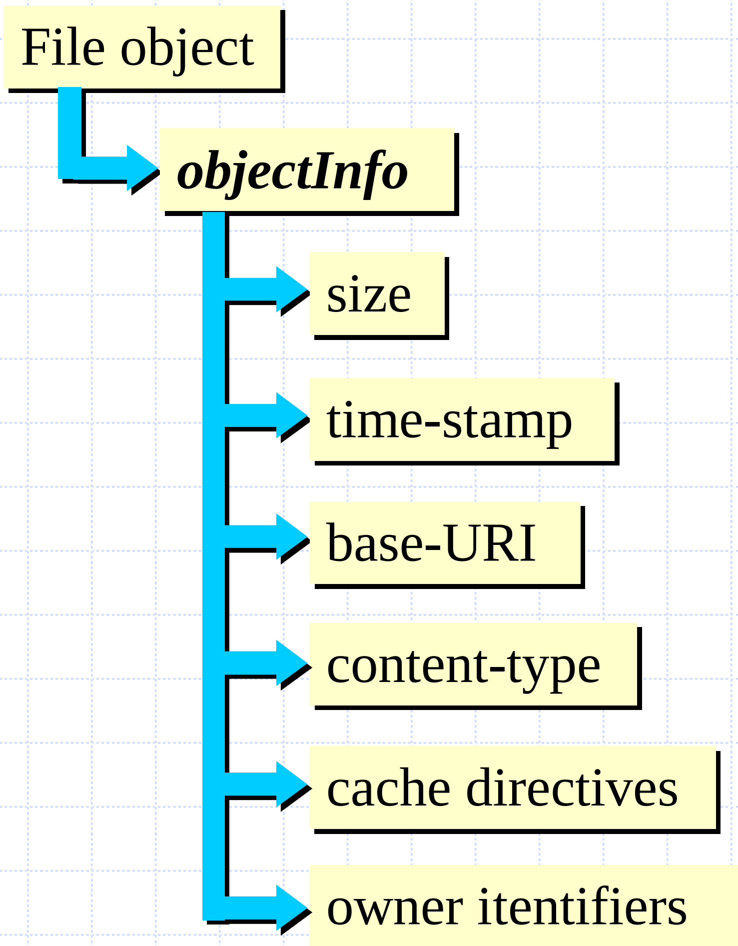
URI transmission (aggregators)

(S13 TSFS work in progress)

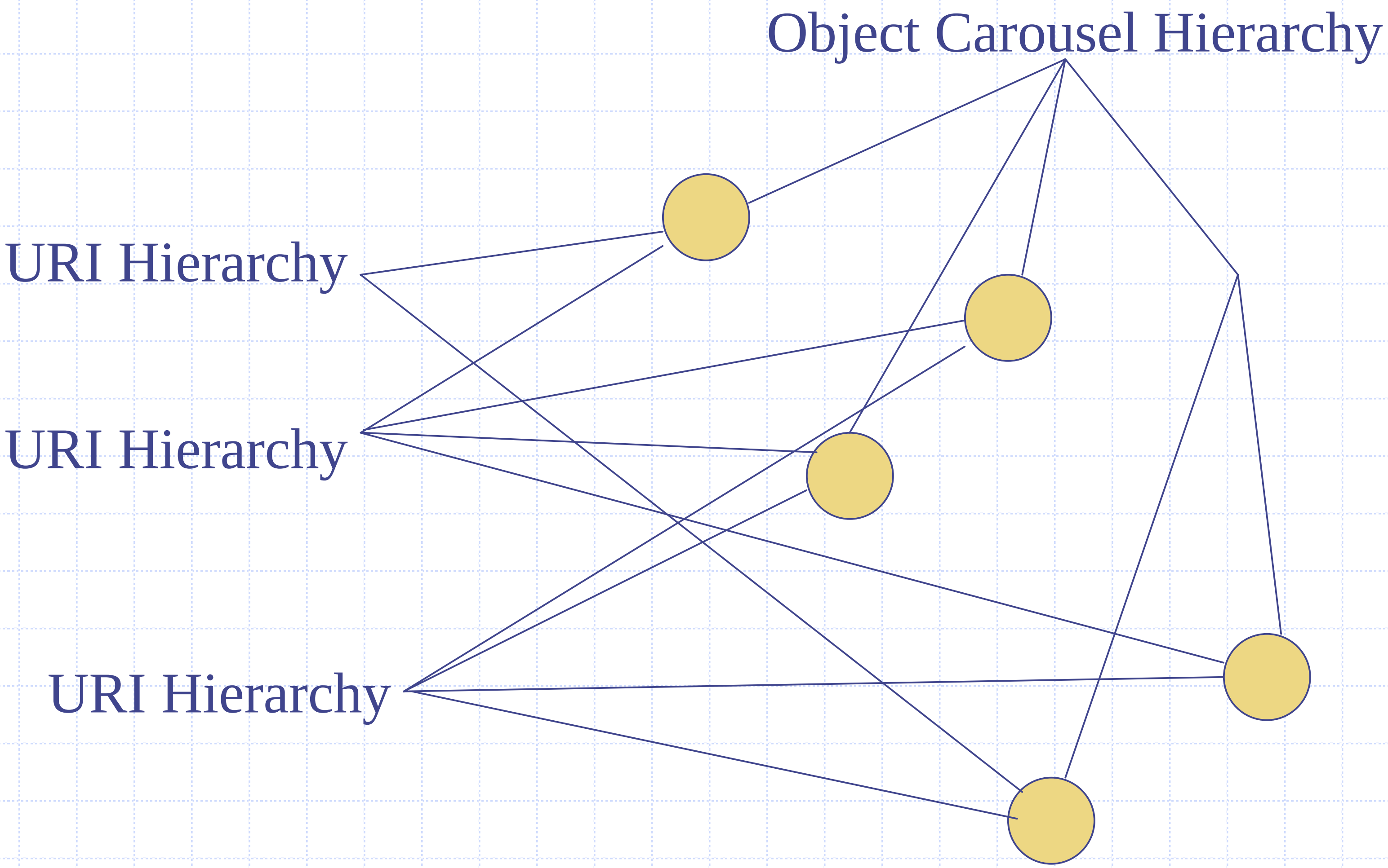
- Directory object
  - Binding structure
    - name
    - ***objectInfo***
      - base-URIs
      - cache directives
      - owner identifiers
      - access attributes

URI transmission (aggregators)

(S13 TSFS work in progress)



Decoupling of Name Spaces



URI resolution (receivers)

- Input: URI
- Output: TS download parameters –
  carousel_id, transaction_id, module_id, version, object_key
- Requires traversing the directory objects
- Can be done completely ‘under-the-hood’ of the TSFS
  - avoid using transport-access API

Application Root Resource

- Contains meta-data about the application
  - Info about authors
  - Info required by execution environment
  - Capability hints
  - Functionality hints
  - Cache information
  - Certificates & security info (future)
  - PVR hints

Capability Hints

- CPU class
- IWS size
- Cache size
- # Decoders
- # Tuners
- Graphic Resolution
- Color Space
- Printing
- Mix requirements
- Input devices
- Buffer models
- Return channel
- Security
- Web-Access
- Financial Transactions
- Export

Functionality Hints

- change minor channel
- change major channel
- active object execution
- scripting execution
- display active area
- display inactive area
- video scaling
- audio mix/alteration
- user/config data read
- user/config data write
- return channel
- printing & other export

Preparation of Meta-Data

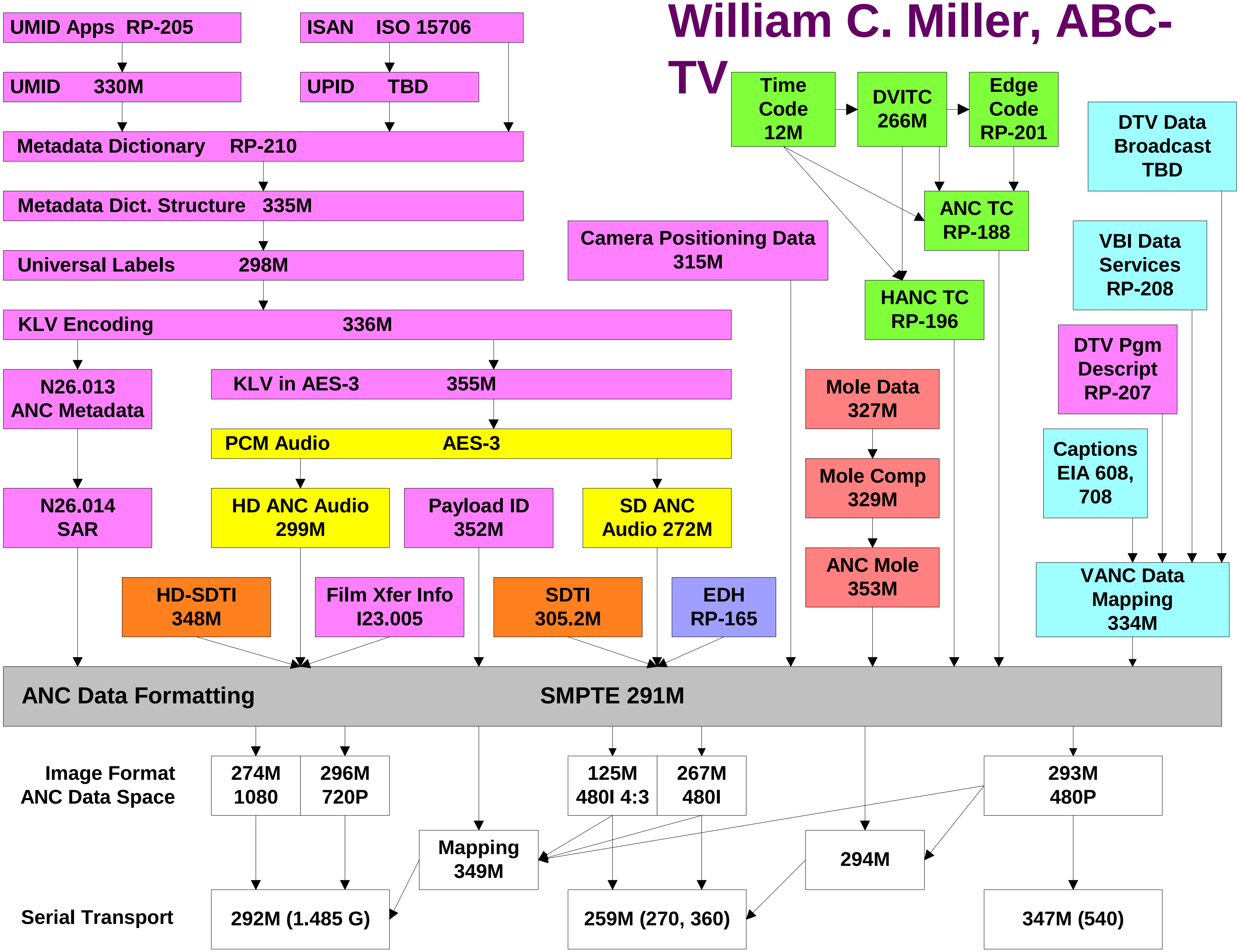
- Distributor Information
- Program-Related
- Video-Related
- Audio-Related
- Data-Related

Data Service Meta Data

- Data Service Announcement (EPG)
- Data Service Signaling
- Application Signaling (Root Resource, Hints)
- Tap Signaling
- Carousel Signaling
- Encapsulation Signaling

ANC Data, Metadata and Data Essence Standards

William C. Miller, ABC-TV



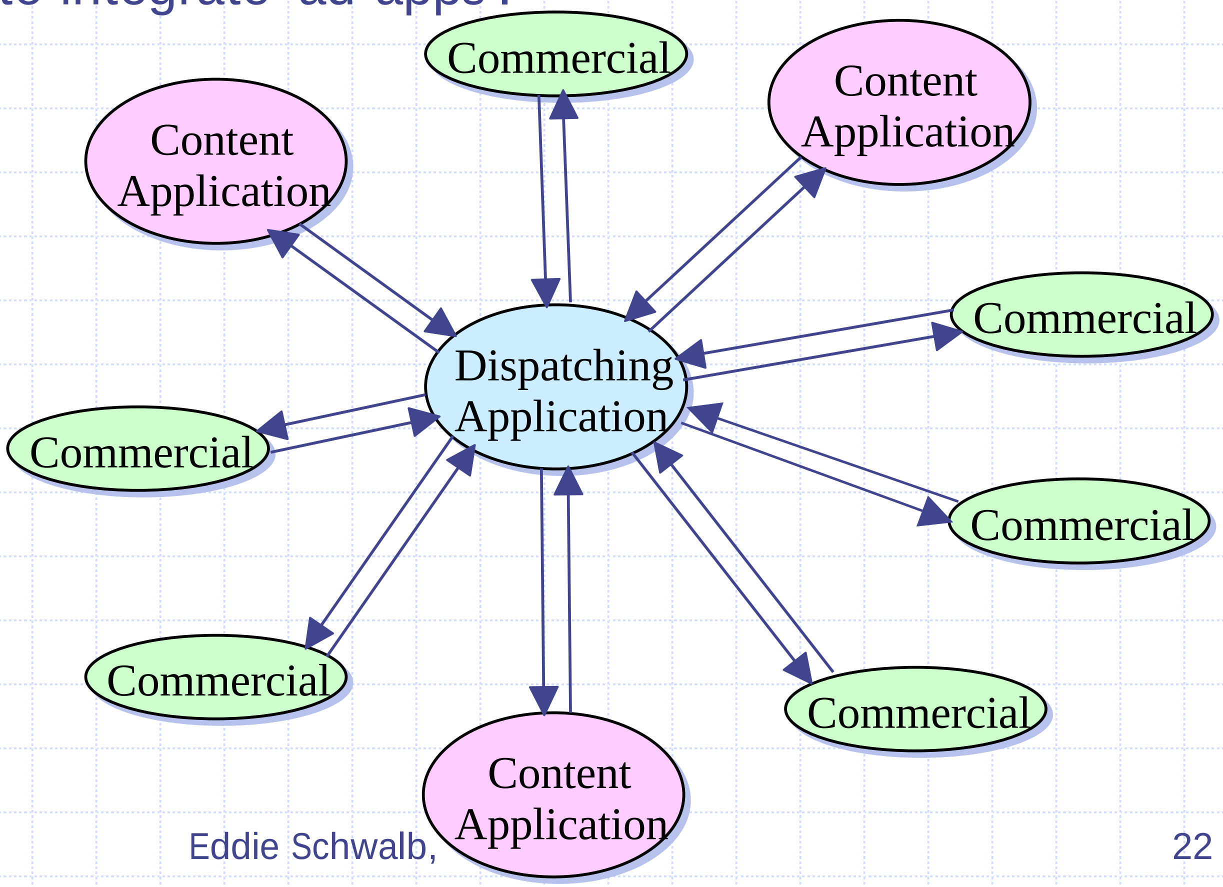
Resource Description Framework (RDF)

http://www.w3.org/TR/REC-rdf-syntax/

- Instead of asking machines to understand people, we ask people to provide information that machines can use understand in order to achieve automation
- Generic end-to-end meta-data solution
  - Content description
    - **title, copyright, reviews, etc.**
  - Workflow annotations
    - **Each step-wise relation has its own**
      - **name space, format, security, etc.**
    - **Supports check-lists, containers, alternatives, statements about statements**.
- Requires one additional small file transmitted with the application

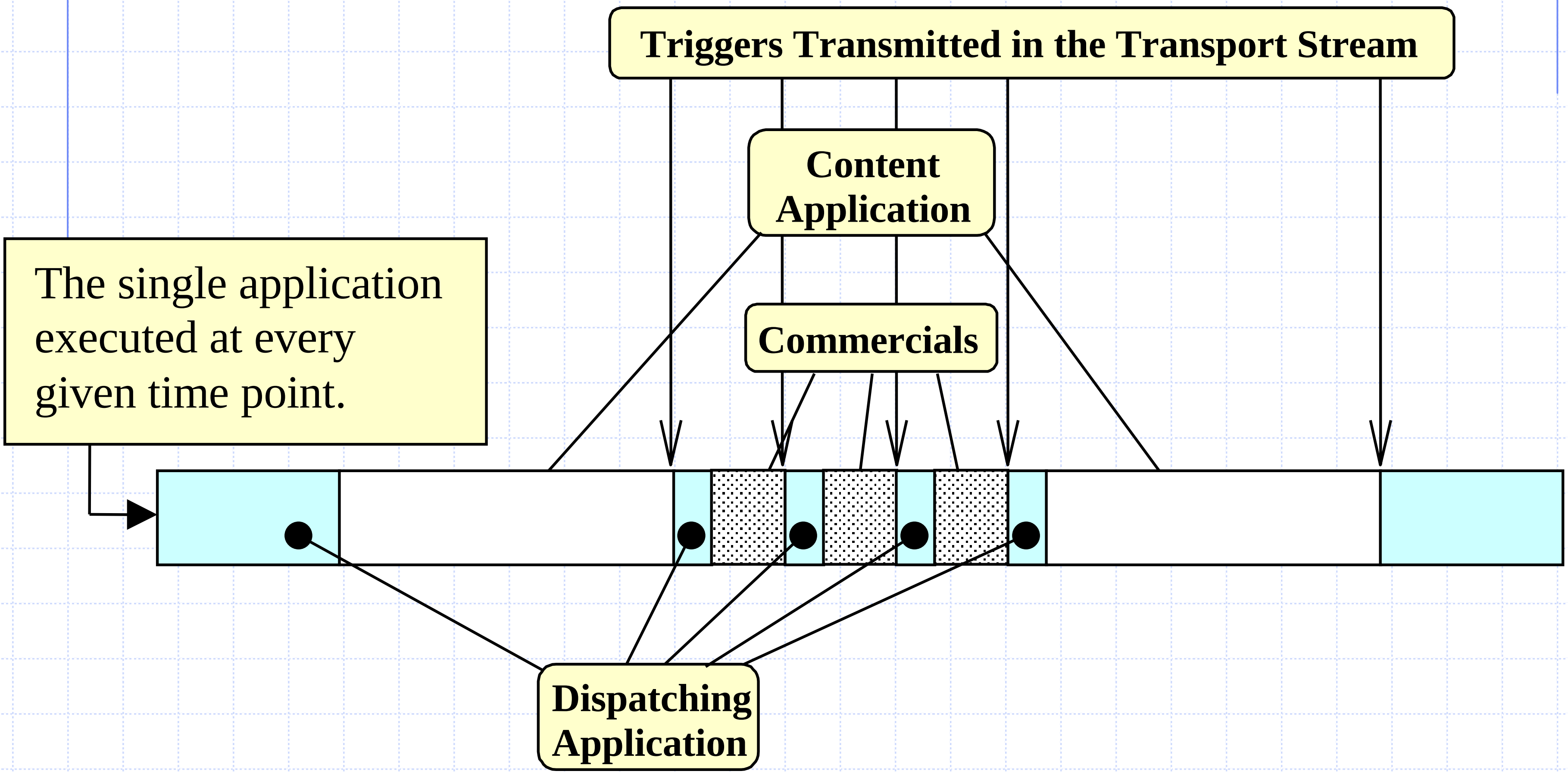
Dispatchers (aggregators)

- Generate menus
- May be generated using ‘story-boards’ or scripts
- Can be used to integrate ‘ad-apps’.



Application Replacement

(DA dispatches DAs, PAs)



Bandwidth Requirements (broadcasters)

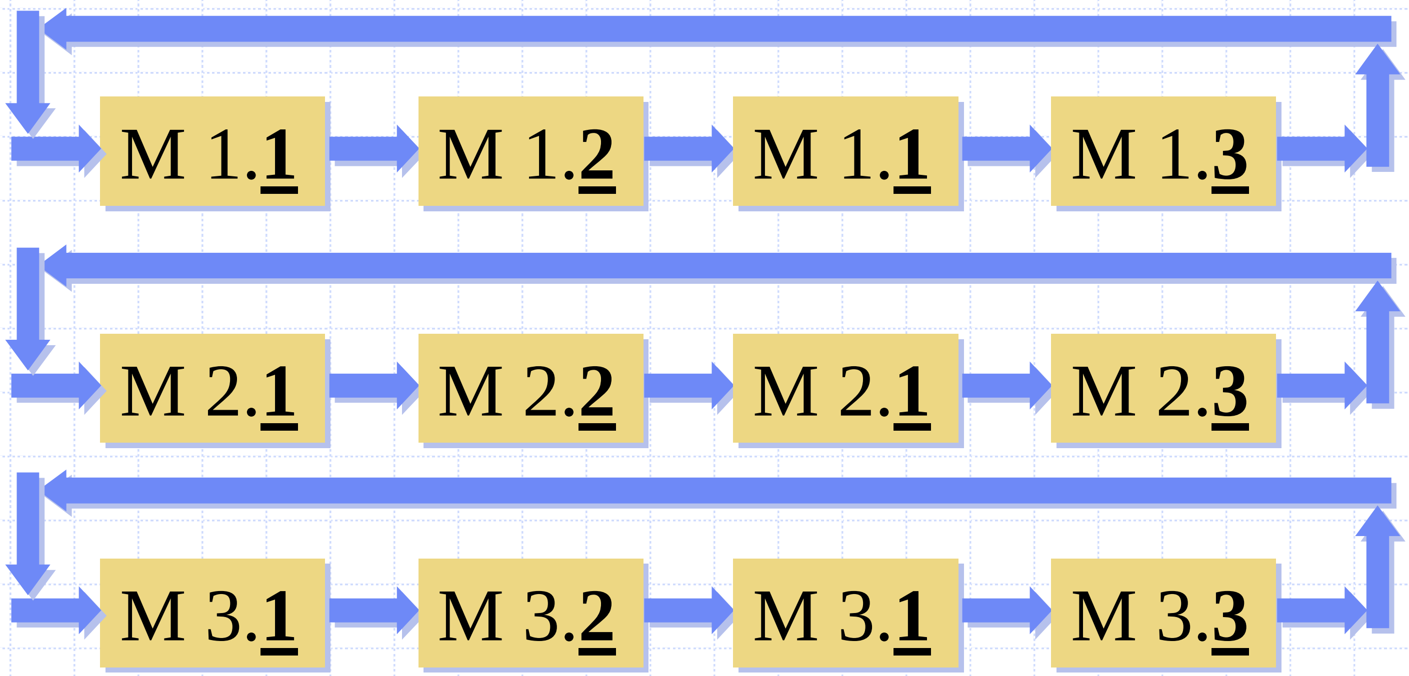
- The total transport stream bandwidth is 19.2mbps.
- Each Data Service Table (DST) is transmitted twice per second.
- Applications should start up within 3 seconds of channel selection.
- Applications should start up within 3 seconds of channel selection.
- Application should respond to selection within 5 seconds thereafter.
- No data should require more than 10 minutes to download, unless done overnight.
- All DSTs and NRTs are allowed up to 6.66% overhead on top of data bandwidth.

Bandwidth Assumptions (broadcasters)

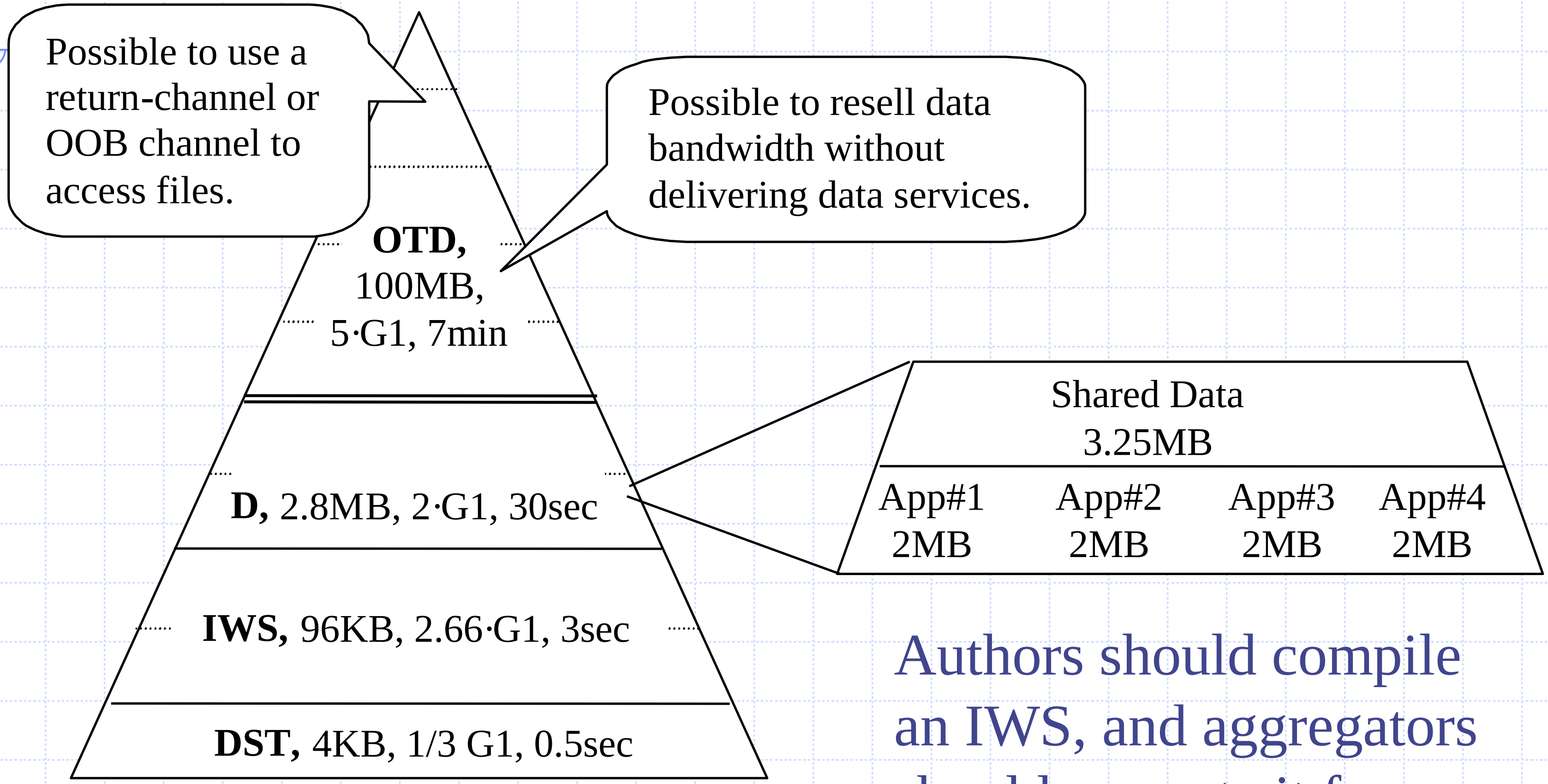
- About 10% of bandwidth will be allocated to data services, i.e., about 5·G1.
- All DSTs and NRTs are allocated 128kbps (=1/3 G1), =6.66% of 5·G1 overhead.
- At most 4 data services will be aggregated within a single transport stream.
- At most 4 application will be aggregated within a single transport stream.
- On the average, first user selection is about 10 seconds after application startup.

Reducing Carousel Average Access Time

Simple technique for
reducing maximum download period 25%



Bandwidth Pyramid (authors, aggregators)

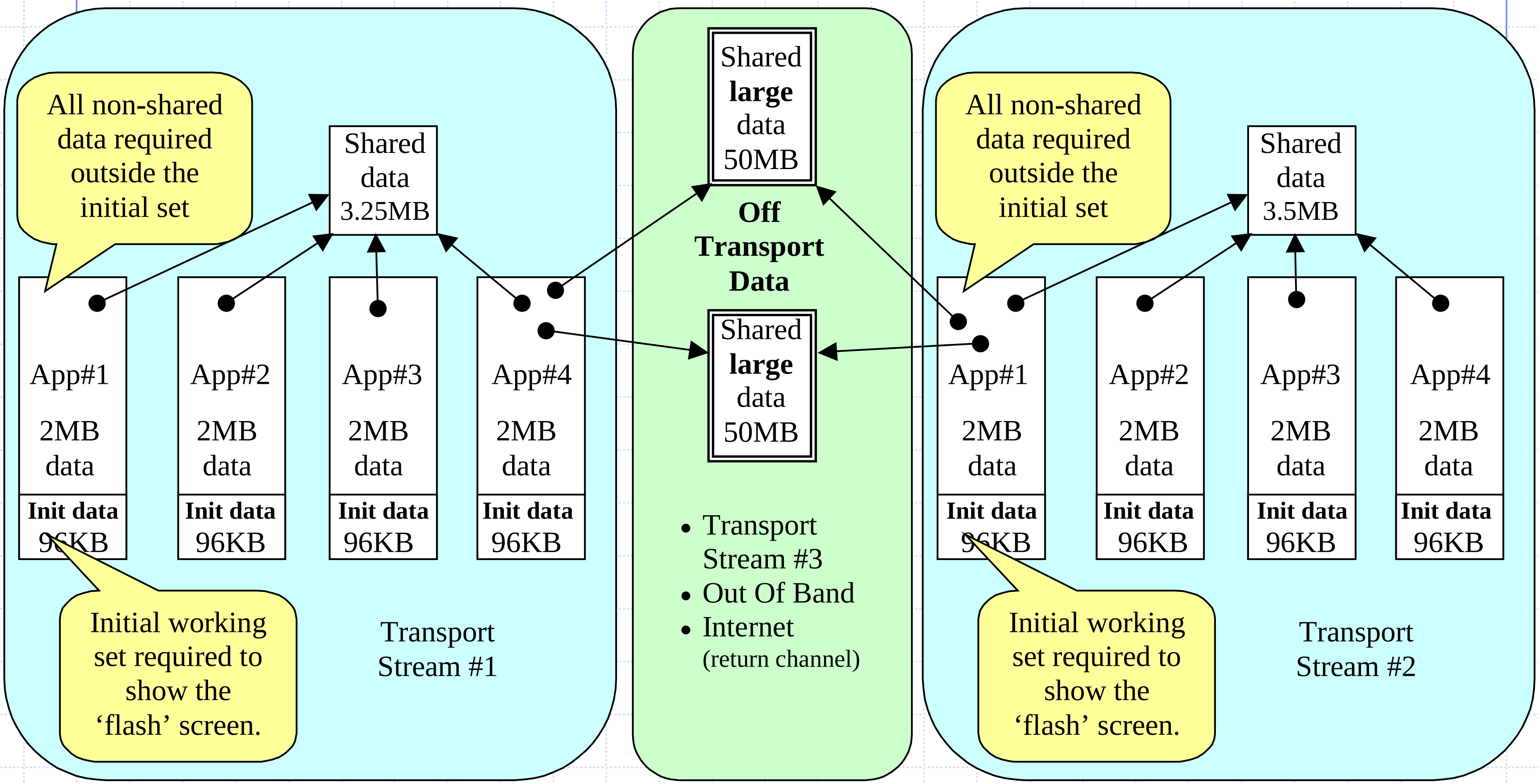


Legend: data_set, size, bandwidth, repeat duration.

DST = Data Service Table
IWS = Initial Working Set
D = Data (or code or both)
OTD = Off Transport Data (Internet)

Authors should compile an IWS, and aggregators should separate it from other data

Transport Layer Configurations



Summary

- Touched on a wide-range of issues:
  - Aggregation model
  - URI mapping
  - Meta-data usage
  - Bandwidth considerations
- Aggregating applications requires numerous types of expertise
- Various types of processing of content may be required

Integration of a RETE-Based Rule Engine into the DASE Environment

Pourya M. Dehnadi

Apex Logic, Inc.
Washington DC, 20007
pourya@apexlogic.com

Much of the operations of a set-top device are dependent on the data that it receives from streams such as the ATSC input stream. The procedural logic residing on the set top device is then able to access the data, perform certain operations, and if appropriate, notify the viewer of any consequences. This data can include PSIP information as well as data that are specific to an in-band or out-of-band application. Due to the data-driven nature of the interactions that can take place between a viewer and a set-top device the integration of a RETE based rule engine can increase performance and provide flexibility for the viewer and the broadcaster. A RETE based rule engine uses a working memory and a set of rules. Each rule is in the form of a material implication. These rules are contained in a rule set, however they are unordered, therefore the correctness of one rule will not impact the performance of other rules. Each viewer can then, through an interface, create a personalized viewing experience based on the rule set that she has created. Multiple rule sets can exist for multiple members of a household. Rule sets can also be uploaded or downloaded allowing for mobility.

Rule sets can be authored, packaged, and transmitted to viewers by the broadcaster. This allows the broadcaster the ability to leverage logic without having to write procedural code, since each rule can be viewed as a declarative unit of operation. The rules can be generated by the broadcaster and downloaded onto the set top devices by the MSO. Since the nature of interactivity over DTV is a push model, as broadcast data is asserted into the working memory of the rule engine, immediate action is taken to fire rules that have antecedents that match current data. Therefore PSIP, in-band, and out-of-band data alike can trigger actions to be performed by the set top device in real time.

The rule engine continually monitors the working memory (by means of the RETE algorithm) for elements that match the antecedents of the rules. If a match is made then the rule engine fires the imperative stated in the consequent of the matched rule. That imperative can be either the assertion of more data into working memory (in the case of forward chaining) or it can be any Java API. By being able to invoke any Java API the entire API's exposed to applications (such as DASE applications) are also exposed to the rule engine.

From an architectural standpoint, the rule engine is integrated into the DASE environment as a DASE-compliant application. This provides for loose coupling between the application and the DASE reference architecture and simplifies implementation. At an implementation level the compiled libraries are relatively small and require minimal memory to run.

Future work will include the development of a parameterized rule engine for the set top box as well as an authoring tool to enable content developers to generate rules in a declarative mode.

Integration of a RETE-based Rules Engine into DASE

Pourya M. Dehnadi

DASE 2001 Symposium

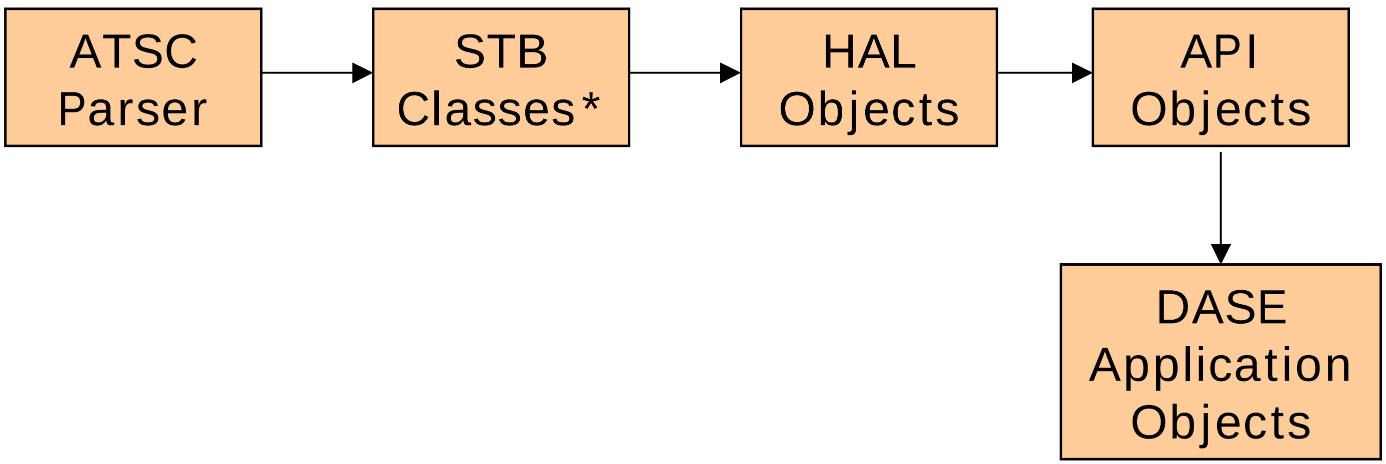
Overview

- Data Overview
- Rules: Using Data for Control
- RETE algorithm and Rules Engines
- Applications
- Authoring and Distribution
- Integration with the NIST RI
- Next Steps
- Conclusions

Data Capture

- ATSC stream
  - Parsed into 3 Components
    - Video – routed to DTV
    - Audio – routed to DTV
    - Data – routed to HAL
- DASE is concerned with the DATA portion of the ATSC stream

Data Flow



* In the case of NIST RI

Kinds of Data

- PSIP Information
  - Program type: Movie, News, Sports, etc.
  - Ratings: PG, R, TV-14, etc.
  - Content: Actor's name, Summary, etc.
- In-band Applications
  - Sport Surveys
- Out-of-band Applications
  - Messenger
  - Stock Ticker

Rules and DASE: Data as Control

- Rules will allow us to leverage the data that we are already collecting, to invoke actions, thus obtaining control.

- Rules can be created by content developers, broadcasters, and viewers.

Brief History of Rules

‘70s	Charles Forgy publishes the RETE algorithm.
‘81	John McDermott and DEC develop R1/XCON, saving DEC millions in manufacturing.
‘83 – ‘90	“Gang of Four” companies, destined to succeed – BusinessWeek; NASA develops CLIPS.
Early ‘90s	Rules have failed, deemed to be expensive to run and maintain.
Mid ‘90s	Distributed Computing, Cheaper RAM, faster processors spark life in Rules.
Today	Rules have resurfaced; Forgy has built RETE2, 100 times faster than the original.

What are Rules?

- Rules are Material Implications
  - "if <antecedent> then <consequent>"
  - The antecedent is a declarative sentence
  - The consequent is an imperative sentence

A rule:

If it is raining, then bring an umbrella.

Not a rule:

If it is raining, then it is overcast.

Rule Engine

Components of a rule engine:

- Rules and rule sets
- Working memory
- Conflict sets

"Instantiation"

- A set of rules and the data in working memory that satisfy these rules

"Conflict set"

- Set of all instantiations at a given point in time

RETE Algorithm

- Uses a Directed Acyclic Graph to create a network that is used to examine the data in working memory against the antecedents of the rules in the rule sets.
- Increases performance, but requires RAM.
- RETE means Network in Latin

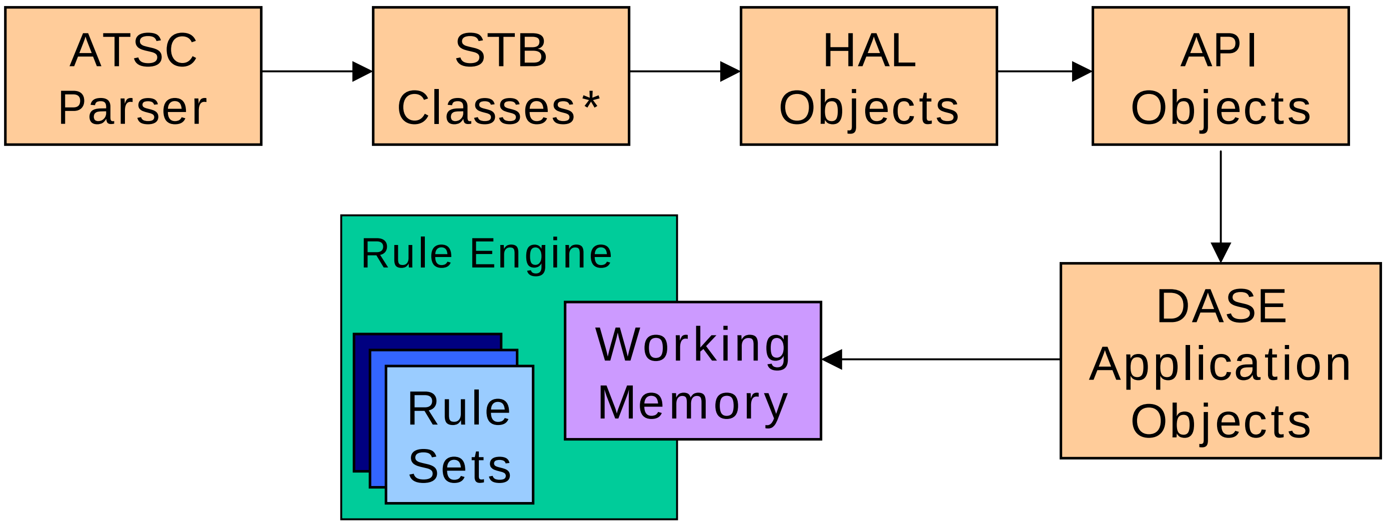
RETE Algorithm

1. Facts (objects and data) are asserted into Working Memory (WME).
2. Rule Engine monitors WME for data that match the antecedent of rules.
3. Instantiations are created.
4. More than one instantiation, results in a conflict set.
5. Conflict resolution is performed.
6. The remaining instantiation is used to “fire” the consequent of the rules.

Benefits of Rules and RETE

- Unordered rules.
- Independence of rules from each other.
- Faster Execution.
- Small units of logic.
- Procedures can be created through forward chaining.
- Backward chaining allows for goal-driven approach... like SQL.

Data Flow, revisited



* In the case of NIST RI

Applications

- Viewer Developed Rules
  - Parental control:
    - If rating is TV-MA or R, then switch to Nickelodeon.
  - Personalized subsets of EPG:
    - Display only sports shows or movies starring Sean Connery.

      (If the program is a movie starring Sean Connery or a sports show, then display it.)

Applications

- Content Provider or Broadcaster Developed
  - In-band:
    - If a viewer has tuned into the middle of the program, then display a summary of what has happened so far.
  - Out-of-band:
    - If the viewer is a sports fan, then show scores on the ticker.
    - If the viewer is an investor, then show stocks on the ticker.
    - Else, show news headlines on the ticker.

Futuristic Applications

- Intelligent media selection based upon viewer profile.
- Example:
  - Viewer Profile indicates male, 28, outdoor activities. The STB could receive two commercials, one for cosmetic products (default) and another for mountain-bikes. This viewer would see the mountain-bike commercial.

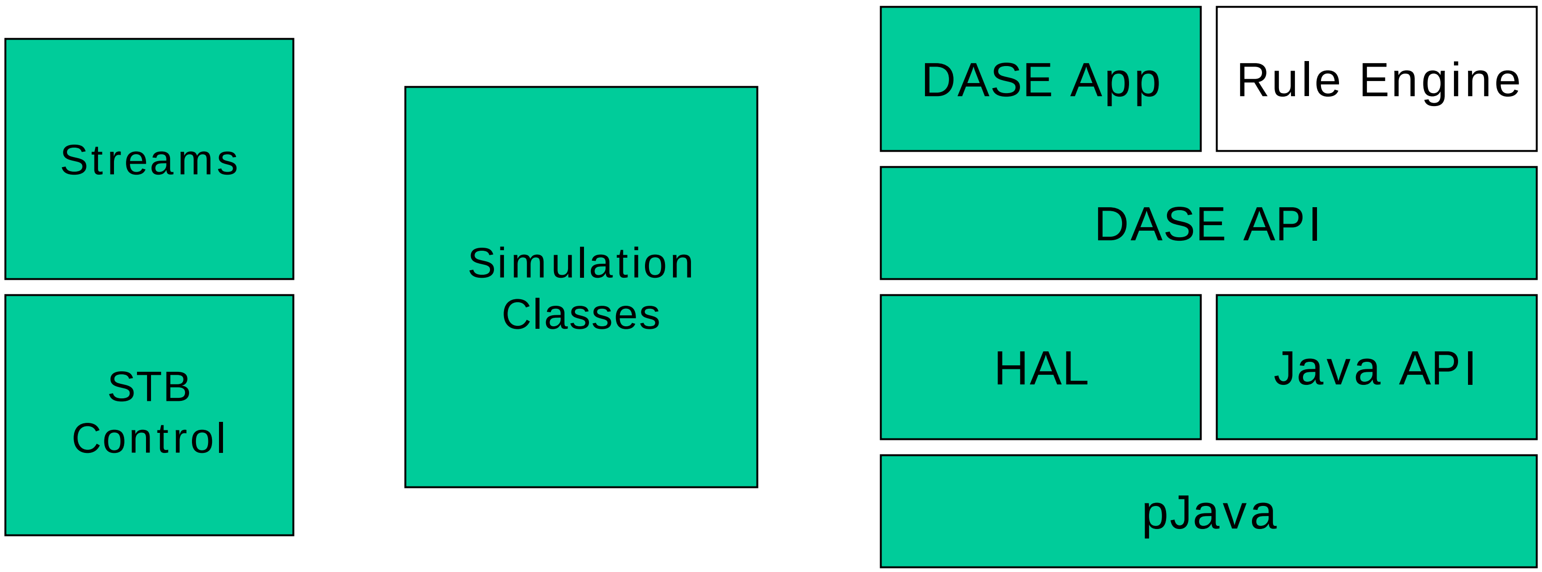
Authoring

- Viewer authoring must be mitigated through a “wizard” like interface that will ensure that rules are correct.
- Content developers may use a more sophisticated “drag and drop” interface with some scripting where necessary.
- Depending on the implementation of the rule engine, rules can be interpreted or compiled.

Distribution

- Applies to Content Developers, Broadcasters, and Viewers.
- Ideally, the rule engine will interpret rules.
- Rule sets can be represented as text-based formats, such as XML.
- Rule sets can be transferred over IP.
- Rule sets can be deployed remotely.

Integration with the NIST RI



The Rule Engine integrates as a DASE application

Next Steps

- Development of an authoring interface.
- Development of a standard file format for distribution.
- Compliance with the privacy issues outlined by the Cable Act and other regulations.
- Optimizations for the set-top box, such as memory footprint (currently, the runtime jar is under 200Kb).

Summary

- Integration of a Rule Engine into DASE allows the use of data for control.
- Viewers, Content Developers, and Broadcaster can author rules given an authoring environment.
- Benefit of rules-based programming over procedural is the decoupling of rules from one another, thus allowing for greater flexibility and availability.
- RETE algorithm provides improved performance over non-RETE rule engines.
- Rule Engine integration with DASE is seamless, since the Rule Engine shall be implemented as a DASE application.