Digital Television for Java™ ME Platform Mobile Devices— Introduction to JSR 272

Antti Rantalahti

Research Manager
Nokia Research Center

Ivan Wong

Distinguished Member
of Technical Staff
Motorola, Inc.

TS-4693

Goal

Learn the basics of Mobile Broadcast Service API for Handhelds. Understand the concepts of mobile digital television and how the API can be used to build rich interactive TV applications.

java.sun.com/javaone/sf

Agenda

Overview

Electronic Service Guide (ESG)

Service and Program Selection

Presentation and Recording

Broadcast Files and Objects

Security and DRM

Purchasing

Status

Agenda

Overview
Electronic Service Guide (ESG)
Service and Program Selection
Presentation and Recording
Broadcast Files and Objects
Security and DRM
Purchasing
Status

java.sun.com/javaone/sf

Expert Group Members

Alticast Corp

Cingular Wireless

Ericsson AB

Gemplus

LG Electronics

Musiwave

Orange France

Samsung

SK Telecom

Telecom Italia

XCE Co.

BenQ

ETRI

Fraunhofer Institute

Hutchison 3G UK

Motorola (spec. lead)

Nokia (spec. lead)

Philips

Siemens

Sprint

Vodafone

Mobile Broadcast Service API for Handheld Terminals—JSR 272

- Application API to give access to mobile digital television technology
  - Agnostic to underlying broadcast technology but…
  - Takes into account the technology specific requirements
- Utilizes existing Java™ ME platform specifications as much as possible
  - Specifies only TV specific functionalities
  - Reuse existing APIs for other general features:
    - Application management and life cycle, purchasing, presentation

Digital Television Concepts

- ESG/EPG—Electronic Service/Program Guide
  - Listing of scheduled broadcast television programs
  - Metadata about the programs; e.g., channel and title
- Program event
  - Primary unit of broadcast programming; i.e., an individual program
- Service
  - Refers to broadcast channel carrying multiple scheduled events
- PVR
  - Personal video recorder

Digital Television Concepts (Cont.)

- File carousel
  - Broadcast stream may carry auxiliary data
  - Contains mainly images and html links
  - Modelled as a read-only file system

- Service-bound application
  - An application that is delivered with the broadcast stream
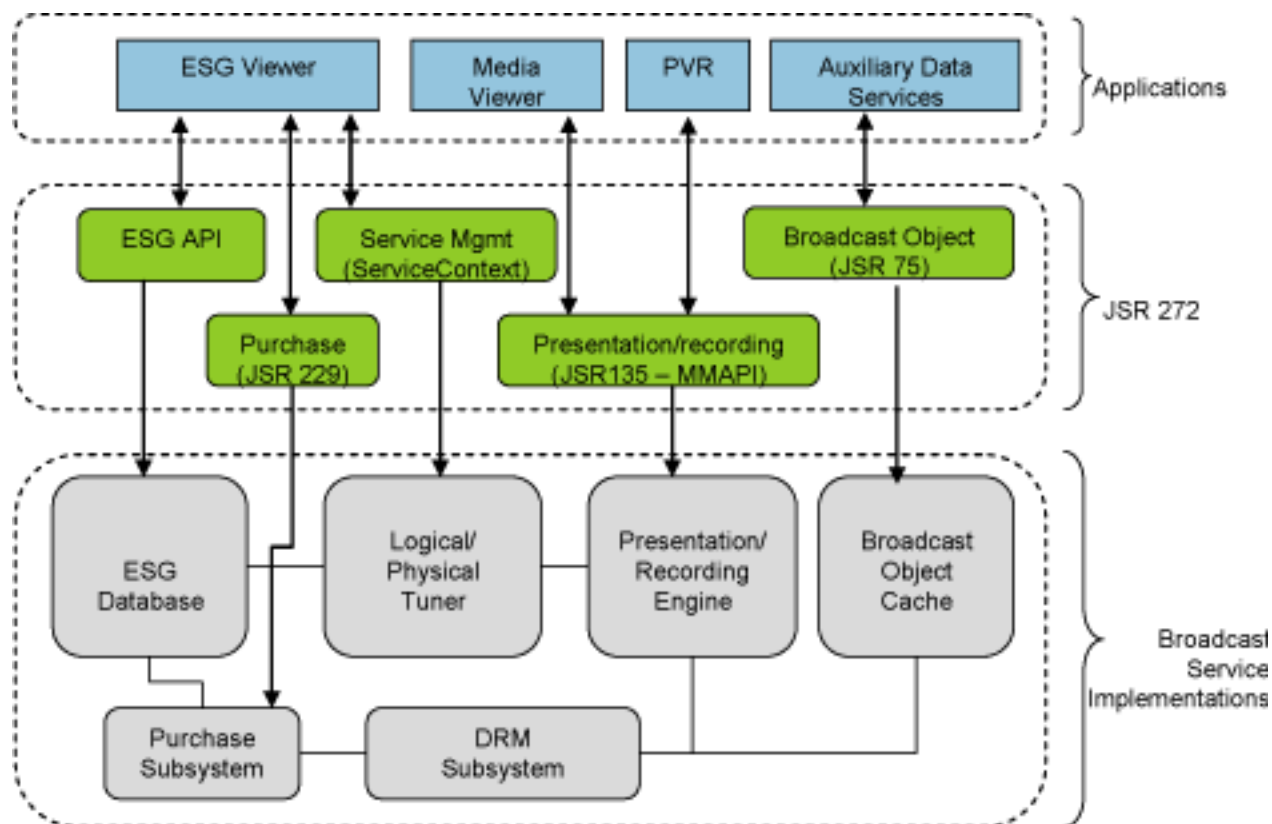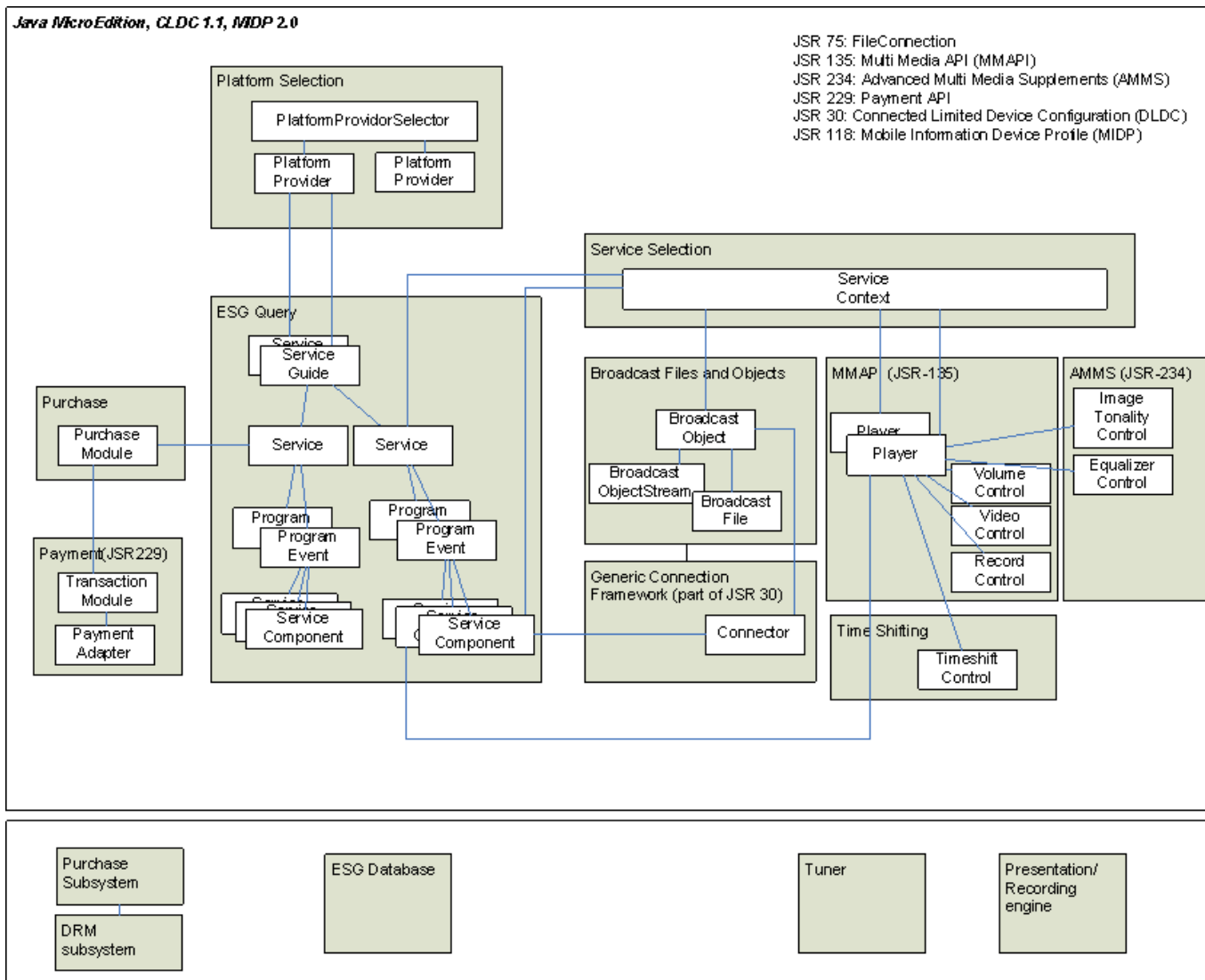 - Typically tailored for the specific content

Main Features

- Access to electronic service guide
- Service and program selection
- Presentation and recording
- Access to broadcast files and objects
- Security and DRM
- Purchasing

JSR 272 Architecture
Architecture of a Typical Implementation

JSR 272 Architecture

java.sun.com/javaone/sf

JSR 272 API Dependencies

- Minimum platform requirements
  - CLDC 1.1
  - Some Java ME profile for the UI: MIPD 1.0, PBP
  - JSR 135 (Mobile Media API) for presentation
- In addition
  - JSR 75: FileConnection to access the broadcast files
  - JSR 229 (Payment API): purchasing
  - JSR 234 (Advanced Multimedia Supplements): additional presentation capabilities

java.sun.com/javaone/sf

Prior Arts

- MHP (Multimedia Home Platform)
  - Targets set top devices with CDC and Personal Basis Profile
  - Defined by DVB and includes other derived standards such as Java based TV, DAVIC etc.
  - Based on the PBP Xlet application model, AWT UI framework and Java based TV which are incompatible with MIDP
- OCAP (OpenCable Application Platform)
  - Extends MHP-GEM (Globally Executable MHP) for the digital cable TV market
- Digital Set Top Profile—"OnRamp to OCAP"(JSR 242)
  - Defines a Java ME profile based on CLDC and a subset of PBP
  - Suitable to run a subset of OCAP on "legacy" set top devices
  - Dependent on AWT UI framework and Java based TV subset (includes Xlet)

Agenda

Overview
Electronic Service Guide (ESG)
Service and Program Selection
Presentation and Recording
Broadcast Files and Objects
Security and DRM
Purchasing
Status

java.sun.com/javaone/sf

Electronic Service Guide

- Program event listing with metadata associated with each event

- Different broadcast specifications specify different schemas for ESG

- JSR 272 abstracts differences in different schemas to provide inter-operability

- ESG modelled as a flattened database

General Broadcast Structure

- Service
  - Broadcast channel containing multiple program events

- Program event
  - Primary unit of broadcast programming
  - For instance movie, music video, etc.

- Service component
  - Media component within a program
  - For instance audio track, video track or language subtitle track

ESG API

- **`Service interface`** abstracts a service
- **`ProgramEvent`** abstracts a program event
- **`ServiceComponent`** describes media components within a service
- **`CommonMetaDataSet`** defines basic attributes common to most broadcast specifications
- **`ESGQueryDatabase`** is access point to database
- **`QueryComposer`** is a factory to compose queries
- **`ESGChangeListener`** to listen for ESG updates

Example: Find All News Programs

```java
// Form the query
Query q = QueryComposer.equivalent
    (CommonMetadataSet.PROGRAM_CONTENT_GENRE, "News");

ProgramEvent programs[];
try {
    // Find the programs
    programs = edb.findPrograms(q);
    for (int i = 0; i < programs.length; i++) {
        // A custom function to display the program info
        displayProgramInfo(programs[i]);
    }
} catch (QueryException e) {}
```

java.sun.com/javaone/sf

More Query Examples

Simple queries:

```
Query q1, q2, q3;

q1 = QueryComposer.equivalent(CommonMetadataSet.SERVICE_NAME, "CNN");
q2 = QueryComposer.after(CommonMetadataSet.PROGRAM_START_TIME,
                                    new Date());
q3 = QueryComposer.equivalent(CommonMetadataSet.PROGRAM_CONTENT_GENRE,
                                    StringAttribute.UNKNOWN_VALUE);
```

Compound queries:

```
Query cq;
cq = QueryComposer.or(QueryComposer.and(q1, q2),
        QueryComposer.equivalent(CommonMetadataSet.PROGRAM_CONTENT_GENRE,
                                    "News"));
```

Agenda

Overview

Electronic Service Guide (ESG)

Service and Program Selection

Presentation and Recording

Broadcast Files and Objects

Security and DRM

Purchasing

Status

java.sun.com/javaone/sf

Service and Program Selection

- Information about services and programs are determined from ESG

- They are selected via Service Selection API
  - Agnostic to bearer technology
  - May require for instance, setting up the radio tuner or joining to IP broadcast group

Service Selection API

- **`ServiceContext`** is the entry point to service selection API

- **`ServiceComponent`** describes media components within a service

- **`ServiceContextListener`** listener for events from **`ServiceContext`**

`ServiceContext` Class

- Environment where a service or program can be selected and presented in a broadcast receiver

- Models the receiver

- State model
  - **PRESENTING**
  - **NOT_PRESENTING**
  - **PRESENTATION_PENDING**
  - **DISPOSED**

`ServiceContext` Class (Cont.)

- Instance management
  - Multiple instances can be created and their resources are centrally managed
  - Application always has access to the default, pre-created instance of `ServiceContext`

- Program or service selection
  - Selecting a program/service results in (MMAPI) players being created for the presentation components
  - Any service-bound application may also be launched

- Component selection
  - Media (audio, video, applications etc.) components can be selected from a broadcast
  - An initial set of components can be set via preference before the program/service is selected
  - After the program/service is selected, components can be added or removed

Agenda

Overview

Electronic Service Guide (ESG)

Service and Program Selection

Presentation and Recording

Broadcast Files and Objects

Security and DRM

Purchasing

Status

java.sun.com/javaone/sf

Presentation and Recording

- Handles the presentation and recording of time-based media such as audio, video and animated content

- Relies on JSR 135 for basic playback and recording; JSR 234 for advanced presentation controls

- Added a new **TimeShiftControl** class for time-shifting

Presentation and Recording API

- JSR 135 `Player` represents a media handler to render and record audio and video

- A Player's `Control` provides the specific media control features like volume control, playback rate, recording

- JSR 234 adds more `Control` classes for advanced media playback features like spatialized audio, equalizer, image tonality control

- `TimeShiftControl` allows the playback to be shifted or delayed to allow the playback to be paused in real-time

- Graphics overlay is supported by allowing the video and GUI rendered on different display planes on the hardware

Example: Selecting a Service and Fetch the Players for Presentation

```
ServiceContext sc =
ServiceContext.getServiceContext(ServiceContext.DEFAULT);

sc.addListener(new ServiceContextListener() {
    public void contextUpdate(ServiceContext s, int event, Object data) {
        if (event == ServiceContextListener.MEDIA_PREPARED) {
            Player players[] = (Player [])data;
              // The players should all be realized at this point.
            for (int i = 0; i < players.length; i++) {
                VideoControl vc =
                    (VideoControl)players[i].getControl("VideoControl");
                if (vc != null) {
                     vc.initDisplayMode(vc.USE_DIRECT_VIDEO,
                         canvas /* a GUI canvas created ahead of time */);
                     vc.setVisible(true);
                } players[i].start();
            }
        }
    }
});
ESGQueryDatabase db = ESGQueryDatabase.getDefaultDatabase();
Service[] current = db.findServices(QueryComposer.CURRENT_PROGRAM);
sc.selectService(current[0]);
```

Agenda

Overview

Electronic Service Guide (ESG)

Service and Program Selection

Presentation and Recording

Broadcast Files and Objects

Security and DRM

Purchasing

Status

java.sun.com/javaone/sf

Broadcast Files and Objects

- Broadcast services may carry auxiliary data complementary to the media stream

- Broadcast objects may be transmitted as a single file, a directory structure with files, or a stream of packet data

- Broadcast objects are received asynchronous to the media streams and may not be immediately available

- Broadcast objects may be updated during the course of the broadcast

Broadcast File and Objects API

- Agnostic to the underlying transport mechanism
- Based on the Generic Connection Framework
- **`BroadcastFile`** abstracts a broadcast object as a file or directory of files
- **`BroadcastObjectStream`** abstracts a broadcast object as a stream of packet data
- **`BroadcastObjectListener`** provides the listener for the asynchronous update events

Example: Broadcast Object

```
public void contextUpdate(ServiceContext s, int event, Object data) {

if (event == ServiceContextListener.SELECTION_COMPLETED) {
    // Retrieve the logo that identifies this program.
    String uri =
s.getService().getStringValue(CommonMetadataSet.PROGRAM_CONTENT_AUX_LOGO);

    if (uri != null) {
      try {
            Connection c = Connector.open(uri);
            if (c instanceof BroadcastFile) {
                // Do something with this logo ...
            }
      } catch (IOException e) { e.printStackTrace(); }
    }
}
```

Agenda

Overview
Electronic Service Guide (ESG)
Service and Program Selection
Presentation and Recording
Broadcast Files and Objects
Security and DRM
Purchasing
Status

java.sun.com/javaone/sf

Security

- Some methods are protected for security
- Permission names are defined for MIPD security framework
- Security for downloaded applications follows the mechanisms of the underlying platform

Digital Rights Management (DRM)

- DRM rights need to be acquired before protected services and program events can be consumed

- Rights can be acquired via different schemes: Conditional Access (CA), OMA DRM (v.1 or 2) etc.

- JSR 272 hides the DRM implementation details but safeguards the applications from violating DRM control at the level of playback, recording and redistribution of the contents

- JSR 272 does not provide any API to query the DRM rights for the contents since these are functionalities common to the platform. It relies on the underlying platform (e.g. MIDP) to provide the functionalities

- Purchase of contents is handled by the payment API

java.sun.com/javaone/sf

Agenda

Overview
Electronic Service Guide (ESG)
Service and Program Selection
Presentation and Recording
Broadcast Files and Objects
Security and DRM
Purchasing
Status

java.sun.com/javaone/sf

Purchasing

- The ESG indicates whether a service/program requires purchase and provides some preliminary pricing information

- JSR 229 Payment API is used to conduct the payment:
    - Identify the payment provider
    - Issue the request to the payment provider
    - Get updated, accurate payment info OTA via the paymentInfo.jpp provisioning
    - Initiate the actual purchase transaction
    - Receive the rights object which contains the decryption key

- A new class, `PurchaseModule` is introduced to adapt the PurchaseItemID (a URI) in the ESG to the PlayFeatureID (an integer) in JSR 229

Purchase Architecture

java.sun.com/javaone/sf
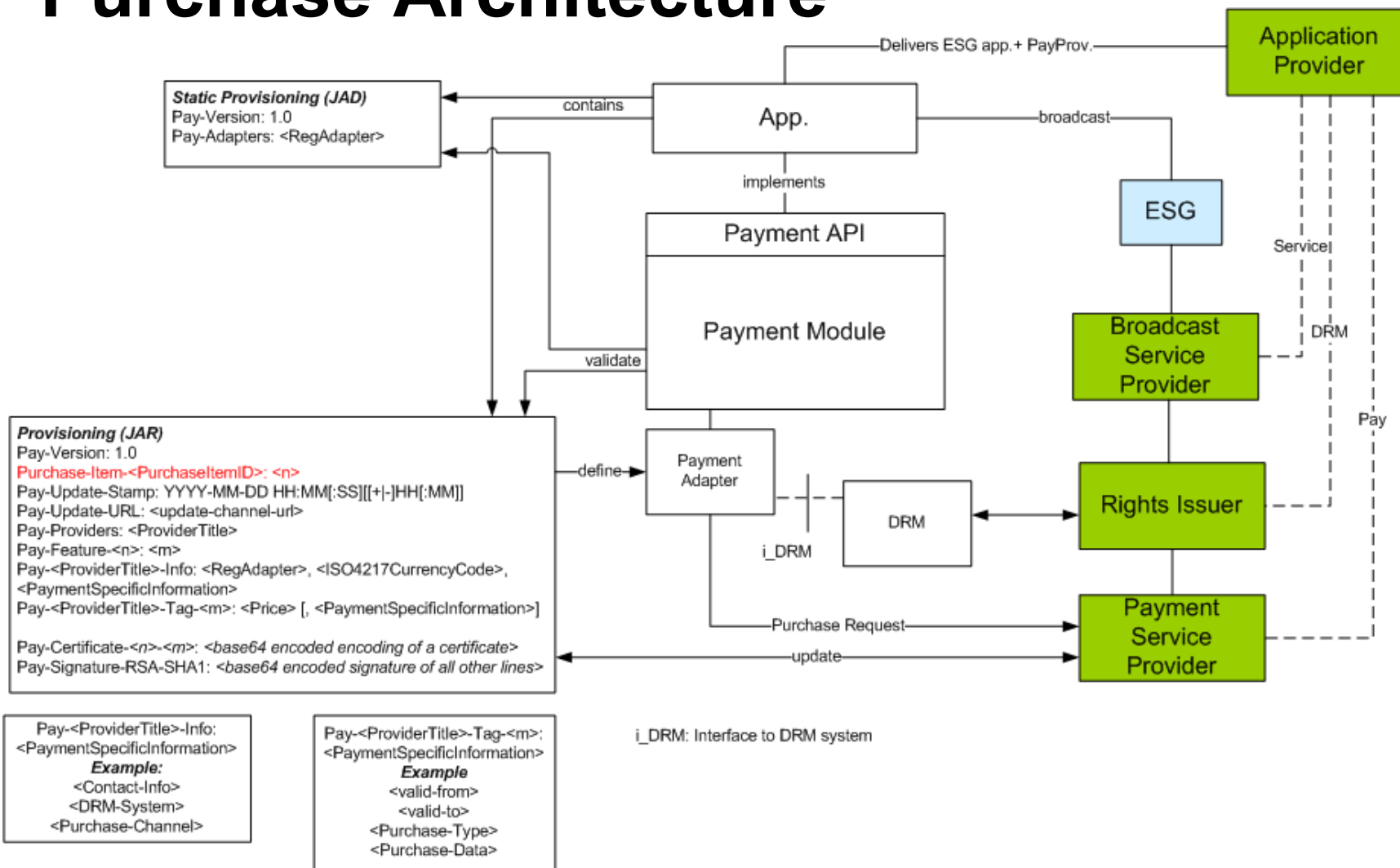
Example: Purchase

```
ProgramEvent p = …;
PurchaseModule purchaseModule = new PurchaseModule(this);

try {
    String id = p.getStringValue(CommonMetadataSet.PROGRAM_ID);
    String title =
            p.getStringValue(CommonMetadataSet.PROGRAM_NAME);
    String description =
        p.getStringValue(CommonMetadataSet.PROGRAM_DESCRIPTION);

    if(!isPurchased(id)){ //send a purchase request int
        feature=purchaseModule.getPaymentFeatureID(id);
        purchaseModule.process(feature,title,description);
    }
} catch (Exception e) { }
```

Agenda

Overview

Electronic Service Guide (ESG)

Service and Program Selection

Presentation and Recording

Broadcast Files and Objects

Security and DRM

Purchasing

Status

java.sun.com/javaone/sf

Status

- April, 2005: JSR submission and EG formation
- June 15: 1st F2F EG meeting
- Oct 25: 2nd F2F EG meeting
- Feb 10, 2006: Early draft review
- April 4: 3rd F2F EG meeting
- Late 2006: Final release

Summary

- JSR 272 is an application API to write mobile digital TV applications

- Agnostic to any specific technology

- Allows presentation and recording of the services

- Gives access to ESG and file carousel

- Provide means to purchase services

- Designed to harmonize with the MIDP platform

java.sun.com/javaone/sf

For More Information

- JSR 272 home page:
  - jcp.org/en/jsr/detail?id=272

java.sun.com/javaone/sf

Q&A

java.sun.com/javaone/sf

Digital Television for Java™ ME Platform Mobile Devices—Introduction to JSR 272

Antti Rantalahti

Research Manager
Nokia Research Center

Ivan Wong

Distinguished Member
of Technical Staff
Motorola, Inc.

TS-4693