OCAP Roadmap and Future Services

Frank Sandoval

Director, OCAP Specifications
CableLabs
http://www.opencable.com

TS-5931

java.sun.com/javaone

OCAP Opportunities for Developers
A view to the future of Cable TV

Peer into the future of interactive services on Cable TV

OCAP Specs

Core specifications

- ## OCAP 1.0-based on PersonalJava™ platform variant

  - Supports typical TV apps—EPG, VOD, etc
  - Supports 'bound' associated with TV programming

- ## OCAP 1.1-based on PBP 1.1

  - Adds metrics
  - Diagnostics
  - 'multi-function' capability
  - Digital Program Insertion (DPI)

java.sun.com/javaone

OCAP Specs

Possible future extension specifications

- SIP/IMS support

- HW accelerated graphics

- Cable-PC

- Mobile

java.sun.com/javaone

OCAP Services

Possible future services

- Nationally distributed interactive content
  - Access to cable network services, like VOD, transactions

- New advertising scenarios
  - Interactive—telescoping to VOD, Request for Information (RFI), etc
  - Addressability—insertion in broadcast, VOD, DVR

- User generated applications

- What are your ideas?

OCAP Future Interactive Services on Cable TV

Mark Jensen

Manager, Interactive Television Platforms

Cox Communications
http://www.cox.com

TS-5931

java.sun.com/javaone

Java™ Platform Interactivity in Cable
Today, tomorrow, and beyond

Eliminating the barriers of **developing** and **delivering** exciting based Java Applications to a Cable Television Device

Agenda

Cable TV's Java Platform Middleware
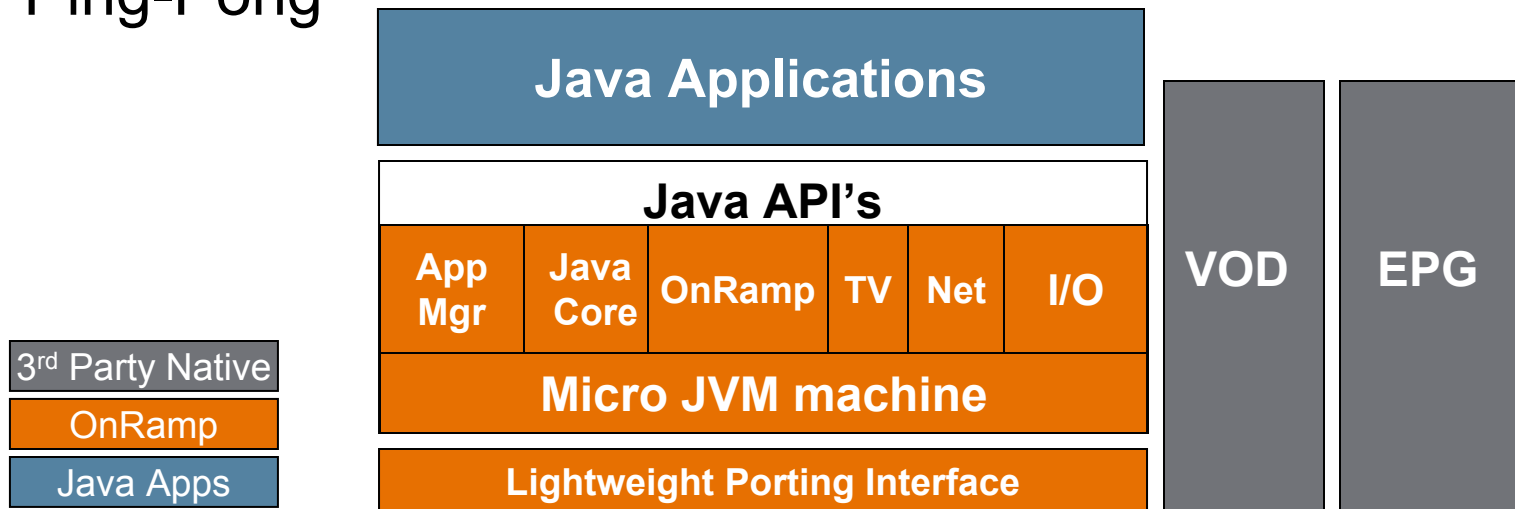
Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

Cable TV's Java Platform Middleware
OnRamp

- Pre OnRamp
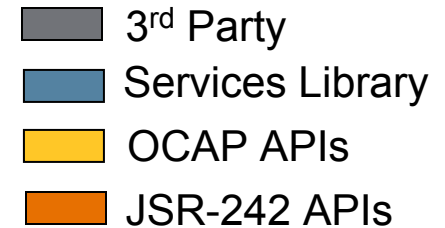 - Micro Java Virtual Machine (JVM™ machine)
  - Ping-Pong

| 3rd Party Native |
| OnRamp |
| Java Apps |

Java Applications

Java API's

| App Mgr | Java Core | OnRamp | TV | Net | I/O |

Micro JVM machine

Lightweight Porting Interface

VOD

EPG

The terms "Java Virtual Machine" and "JVM" mean a Virtual Machine for the Java™ platform.

Cable TV's Java Platform Middleware
OnRamp

- JCP^{SM} Services—JSR 242

- Subset of OCAP

Legend:
- 3rd Party
- Services Library
- OCAP APIs
- JSR-242 APIs

Guide
VOD

Java App

Java App

CLDC 1.0	Core Java	DAVIC	JavaTV	DVB	HAVi	AWT	Crypto	JMF 1.0	OCAP DVR	PVR Eng	DB	VOD	EBIF Engine

Java Virtual Machine (CLDC)

Settop Hardware and O/S
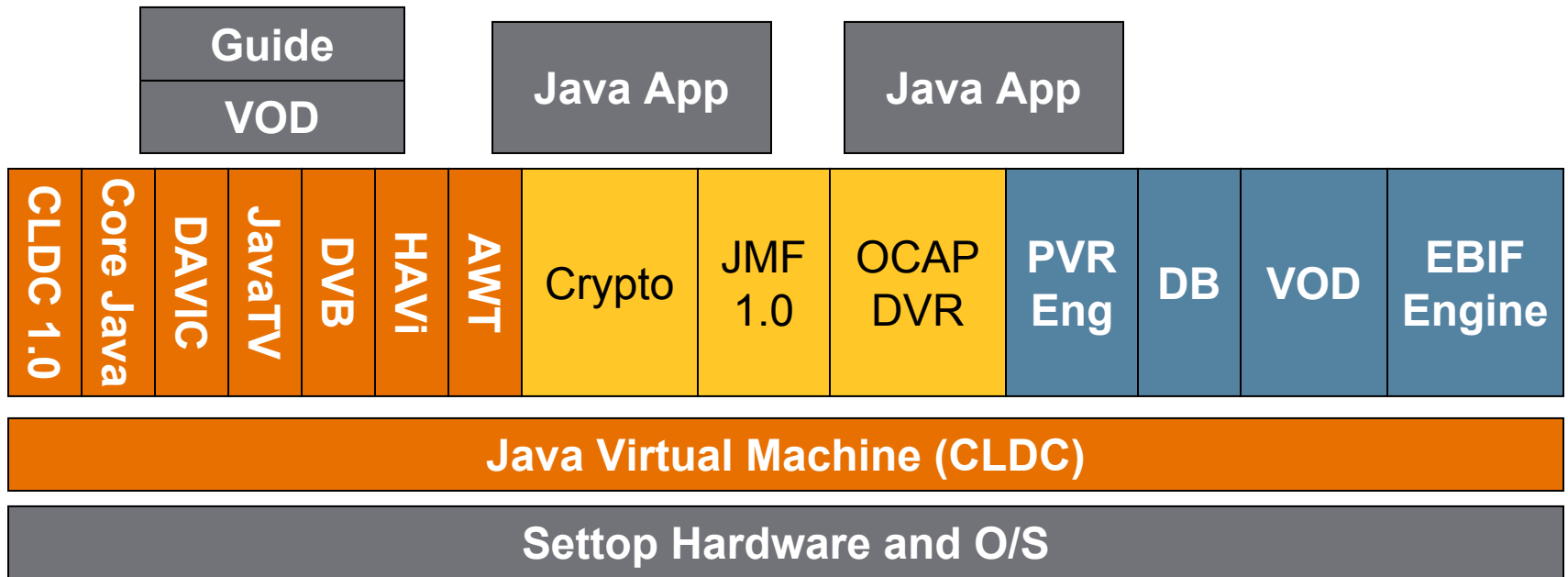
Cable TV's Java Platform Middleware

OCAP

- Based on GEM
- CDC Today
- Moving to PBP
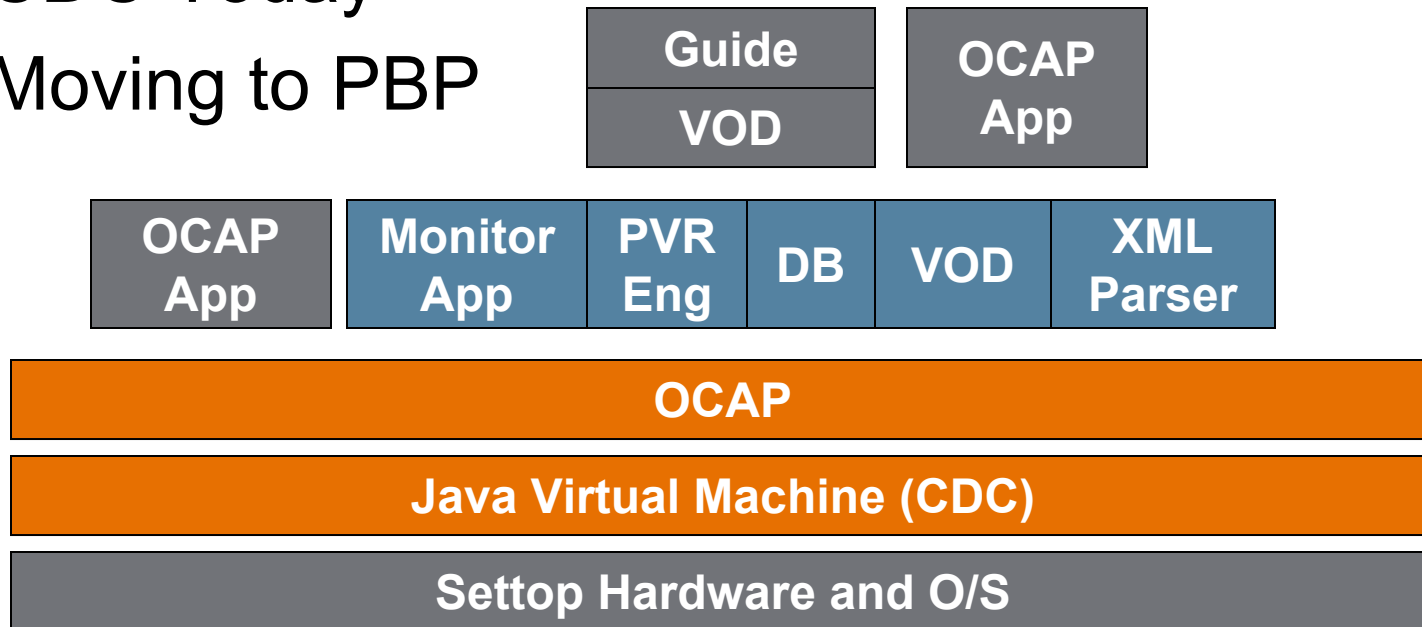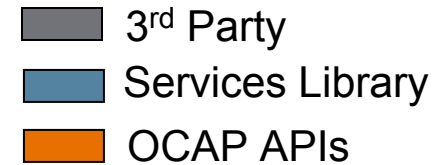
Legend:
- 3rd Party (gray)
- Services Library (blue)
- OCAP APIs (orange)

Guide	OCAP
VOD	App

OCAP App	Monitor App	PVR Eng	DB	VOD	XML Parser

OCAP

Java Virtual Machine (CDC)

Settop Hardware and O/S

Agenda

Cable TV's Java Platform Middleware

Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

Process
Steps to Success

- Develop in a Java Environment
  - Standard IDE Tools

- Simulate with an
  - OCAP Simulator
  - Cable TV Network Simulation

- Participate in OCAP Interops

- Work with an OCAP Device Manufacture

- Contact an MSO

java.sun.com/javaone

Tools
How to get started

- OCAP Emulators Development Tools
  - VisionWorkbench, MHDK

- Network Simulators and Production Tools
  - Ensequence, eMUSE, Tandberg/GoldPocket

- Video and Data Synchronization Tools
  - Object Carousel with RF Modulator to Set-top

- Set-top serial output of a debug device

java.sun.com/javaone

Agenda

Cable TV's Java Platform Middleware

Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

java.sun.com/javaone

Device Types

- Constrained set-top device (CLDC)
  - OnRamp
    - Limited Network Capability (Legacy OOB)

- Advanced set-top devices (CDC)
  - OCAP
  - DOCSIS Cable Modem built right in

- Advanced CableCARD Televisions (CDC)
  - OCAP
  - DOCSIS Cable Modem built right in

- Advanced DCAS Televisions and set-tops (CDC)
  - OCAP
  - DOCSIS Cable Modem built right in

java.sun.com/javaone

Agenda

Cable TV's Java Platform Middleware

Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

java.sun.com/javaone

Cables Transport Network
Limited Capability

- OOB Application and Data Delivery
  - SCTE 65 and 54
  - UDP/IP with limited Bandwidth

- IB Application Delivery
  - Constrained
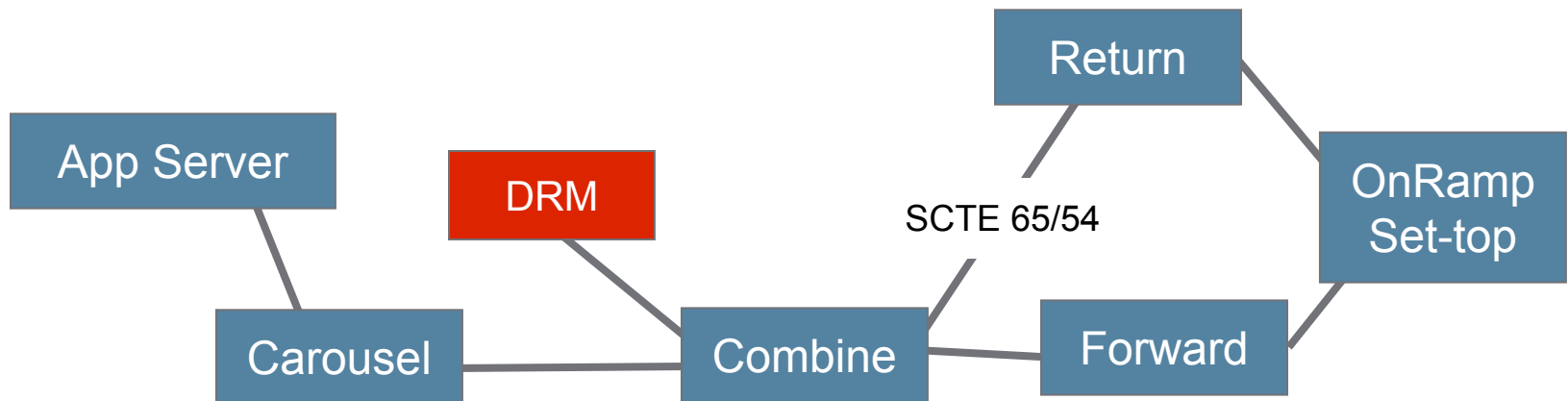
Cables Transport Network
DOCSIS & DSG

- ## DOCSIS
  - ### TCP/IP

- ## DSG
  - ### VPN like tunnels
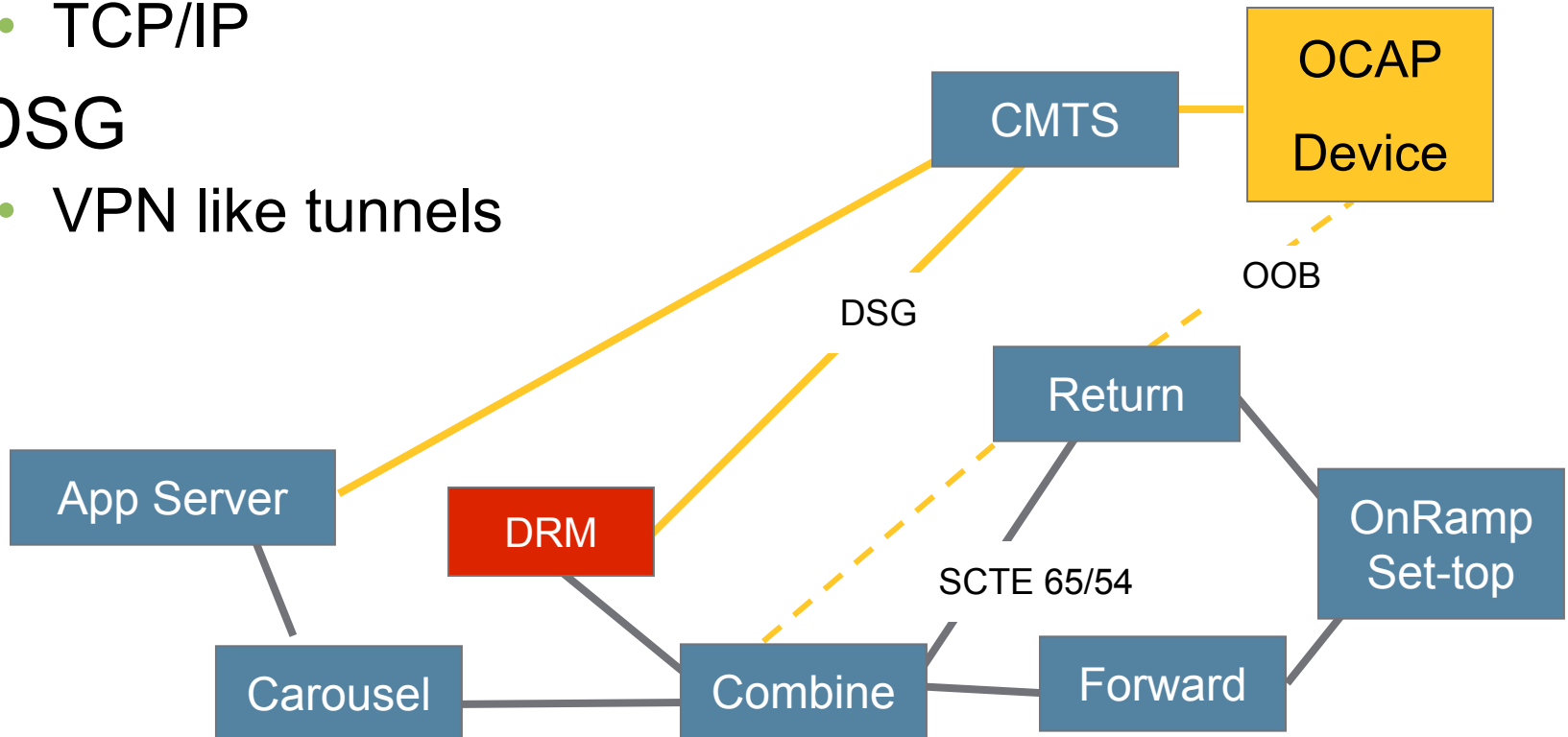
CMTS

OCAP Device

OOB

DSG

Return

App Server

DRM

OnRamp Set-top

SCTE 65/54

Carousel

Combine

Forward

Agenda

Cable TV's Java Platform Middleware

Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

Today's Java Applications

- News and Information
  - Weather
  - News
  - On TV
  - At the Movies
  - Horoscopes
- Advertising
  - Cox Media Advertisers
  - Cox Products and Services

java.sun.com/javaone

News and Information

Advertising

Today's Java Applications

- Customer Care
  - View and Pay Bill
  - Sign up for EasyPay
  - Product FAQ

- Order Services
  - Premium Channels
  - Cox Products and Services

- E-Mail Viewer (Integrated with High Speed Data)
  - Read and Delete E-Mail

java.sun.com/javaone

Customer Care

Order Services

E-Mail Viewer

Tivo and OnRamp

Source: Tivo

Tivo and OnRamp

Source: Tivo

java.sun.com/javaone

Tivo and OnRamp

Source: Tivo

Agenda With Section Highlights

Cable TV's Java Platform Middleware

Tools and Process

Device Types

Network Capabilities

Today's Java Applications

Tomorrow's Java Applications

java.sun.com/javaone

Today's Java Applications

- Targeted Advertising
    - Basic Profile Services

- Enhanced Telephony Integration
    - Caller ID
    - Call Disposition
    - Call Forwarding

- E-Mail Viewer (Integrated with High Speed Data)
    - Read and Delete E-Mail

Targeted Advertising

Enhanced Telephone Integration
Incoming Call with Caller ID

Enhanced Telephone Integration
Disposition Banner

Enhanced Telephone Integration
Most Recent Calls and Call Routing Configuration
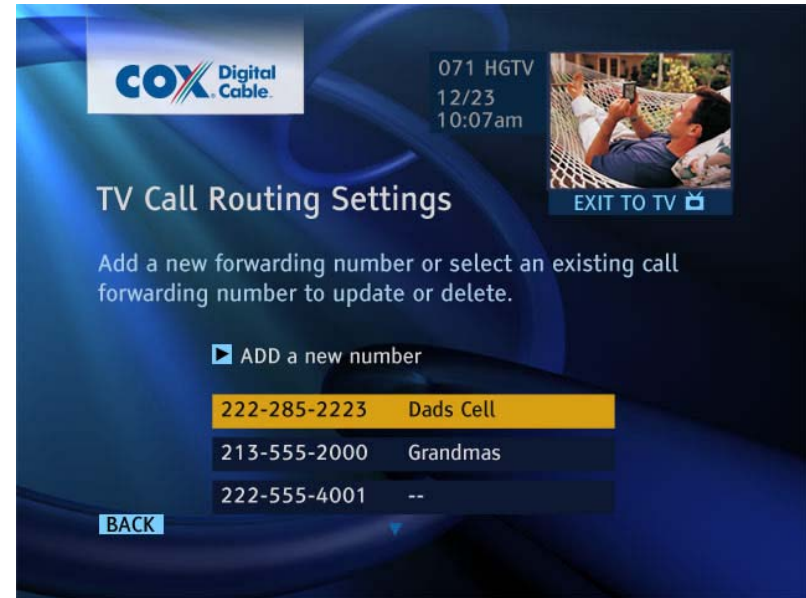
Enhanced Telephone Integration
Disposition Banner with Active Call Forward List

java.sun.com/javaone

Summary

- Java Platform Middleware on set-tops are being deployed

- Java applications are gaining in popularity

- Advertising is gaining momentum

- Development Roadblocks are being Eliminated

- Ubiquitous Acceptance is soon

java.sun.com/javaone

OCAP Future Interactive Services on Cable TV

Michael Lobb

TVWorks

Cox Communications
http://www.cox.com

TS-5931

java.sun.com/javaone

What Is a Library in OCAP?

- OCAP provides the ability to deploy library code via its org.ocap.system.RegisteredApiManager class

- Code registered in this way is then made available to applications in other Java platform contexts

- Each application that wishes to use the code will have its own copy on its system heap, i.e. static variables do not exist across applications

- The library code is effectively added to each application's classpath

java.sun.com/javaone

Why Would I Use This?

- Code sharing within a suite of applications to save footprint

- Code implemented by one company with usage sold to others

- A game pack could have an extended set of widgets that are common to each one

- Providing middleware type APIs as new features or extensions of an existing platform (more on this later)

java.sun.com/javaone

How an Application Uses a Library

- In OCAP applications are signalled with an XAIT which is conceptually similar to a Java Application Descriptor (JAD) file in the mobile space

- One of the XAIT fields is a list of library names an application is allowed to use

- Application must be signed with the same certificate as the library

- The library must be registered before the application is launched or it will not be able to "see" the code

java.sun.com/javaone

RegisteredApiManager

```
static RegisteredApiManager getInstance()
        Gets the singleton instance of the Registered API manager.


abstract java.lang.String[] getNames()
        Gets a list of registered APIs.


abstract java.lang.String[] getUsedNames()
        Gets a list of registered APIs that are in use by the caller.


abstract java.lang.String getVersion(java.lang.String name)
        Gets the version of a registered API, or null if it is not
registered.


abstract void register(java.lang.String name, java.lang.String version,

        java.io.File scdf, short storagePriority)
        Registers an API with the implementation.


abstract void unregister(java.lang.String name)
        Unregisters an API from the implementation.
```

SCDF Sample

```xml
<?xml version="1.0" encoding="UTF-8"?>
<applicationdescription>
  <dir name="com">
    <dir name="tvworks">
      <dir name="package1">
        <file name="BinaryProtocol.class" size="438"/>
        <file name="CommandMessage.class" size="971"/>
        <file name="Element.class" size="2834"/>
      </dir>
      <dir name="package2">
        <file name="ElementVisitor.class" size="147"/>
        <file name="LogMessage.class" size="1519"/>
      </dir>
    </dir>
  </dir>
</applicationdescription>
```

Code Sample

```java
public void registerAnApi() {
    //establish the library's name
    String name = ...
    //establish the library's version number
    String version = ...
    //locate the descriptor file
    File scdf = new File(...);
    //obtain an instance of the RegisteredApiManager
    RegisteredApiManager manager = RegisteredApiManager.getInstance();
    //register the descriptor file
    manager.register(name, version, scdf, (short)255);
}
```

java.sun.com/javaone

Code Sample (Cont.)

```java
public void enumerateApis() {
    //obtain an instance of the RegisteredApiManager
    RegisteredApiManager manager = RegisteredApiManager.getInstance();
    String[] registeredNames = manager.getNames();
    for ( int i=0, length = registeredNames.length; i < length; i++ ) {
        System.out.println(registeredNames[i]);
    }
}
public void upgradeApi(String name) {
    //locate the descriptor file
    File scdf = new File(...);
    //obtain an instance of the RegisteredApiManager
    RegisteredApiManager manager = RegisteredApiManager.getInstance();
    String newVersion = manager.getVersion(name)+".1";
    manager.unregister(name);
    manager.register(name, newVersion, scdf, (short)255);
}
```

java.sun.com/javaone

Model for Services
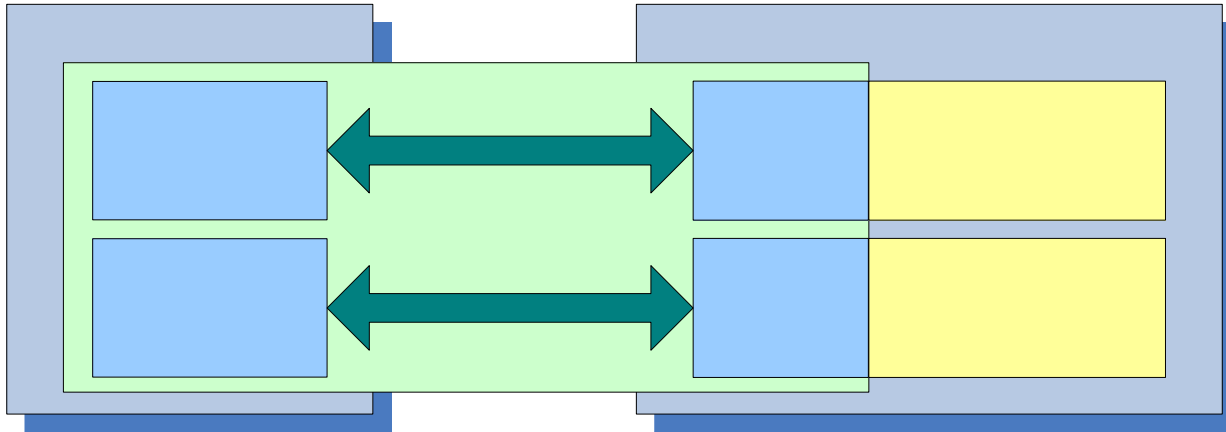
- Each application runs in its own Java execution context; these VM instances do not and can not share memory

- Interprocess communication is handled via the LinkRegistry and the Registered API mechanism

Uses of Library Code

- Program listings database
- Video On Demand—streaming video
- PVR—personal video recorder
- Parental Control storage and enforcement

java.sun.com/javaone

Program Guide Prototype

VOD Category Listing Prototype

VOD Movie Listing Prototype

VOD Movie Information Prototype

VOD Streaming Video Prototype
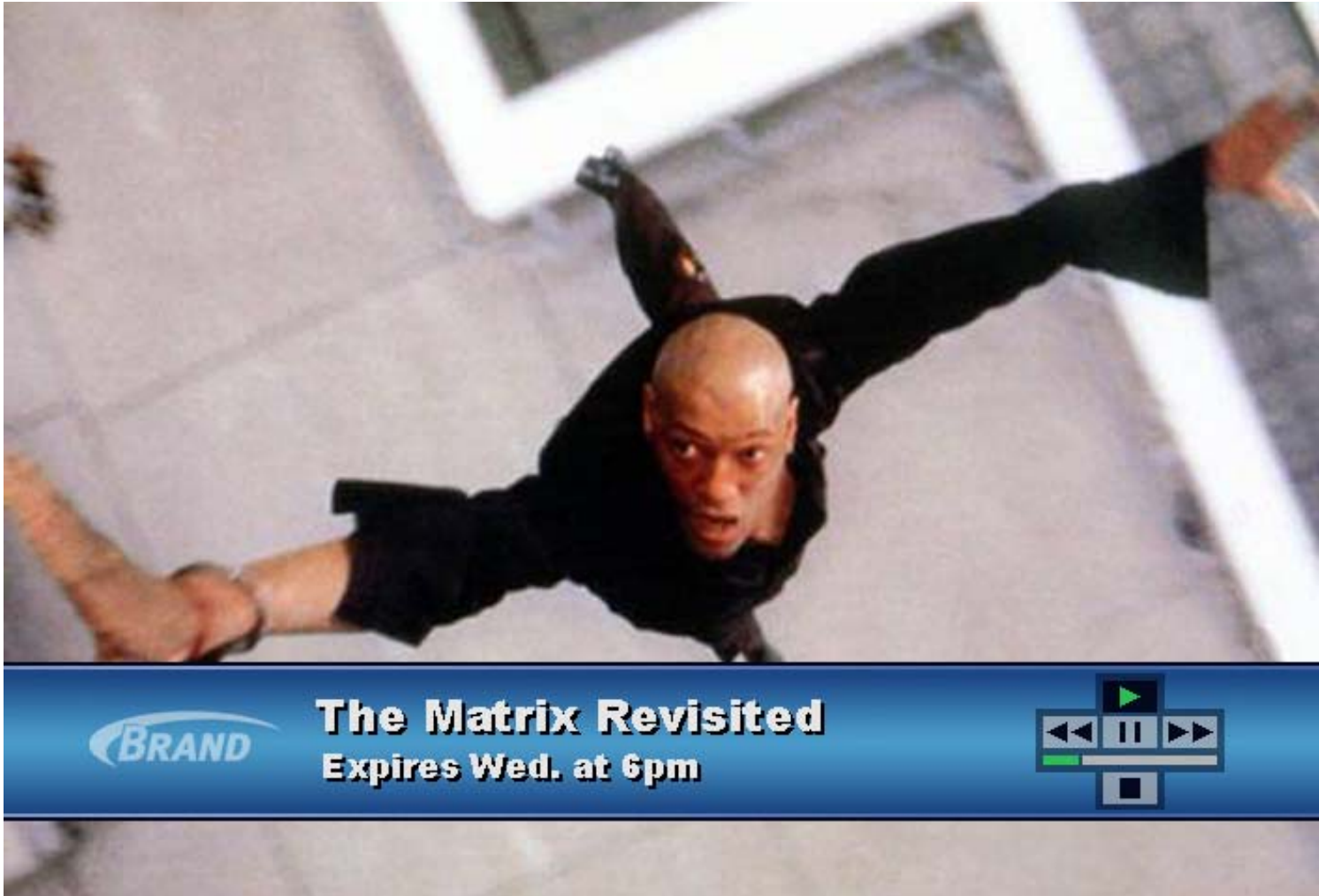
Q&A

OCAP Roadmap and Future Services

Frank Sandoval

Mark Jensen

Michael Lobb

TS-5931

java.sun.com/javaone