OCAP: A Summary of Technical Features and APIs

Kevin Flanagan

Software Engineer
CableLabs
http://www.opencable.com

Labeeb Ismail

VP of Technology
Comcast Cable
http://www.comcast.com

TS-0011

OCAP for Java™ Platform Gurus

A functional understanding of OCAP in 60 minutes

Learn the basics of the OCAP platform **now** to begin developing the interactive television applications of tomorrow.

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

OCAP—Extensions

OCAP Development Example

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

OCAP—Extensions

OCAP Development Example

CableLabs Overview

- CL is a non-profit technical consortium
  - Members are cable network operators
- Various projects within CL
  - DOCSIS, PacketCable, OpenCable
- Projects are driven by member business needs
- Vendors and members participate in projects

OpenCable Overview

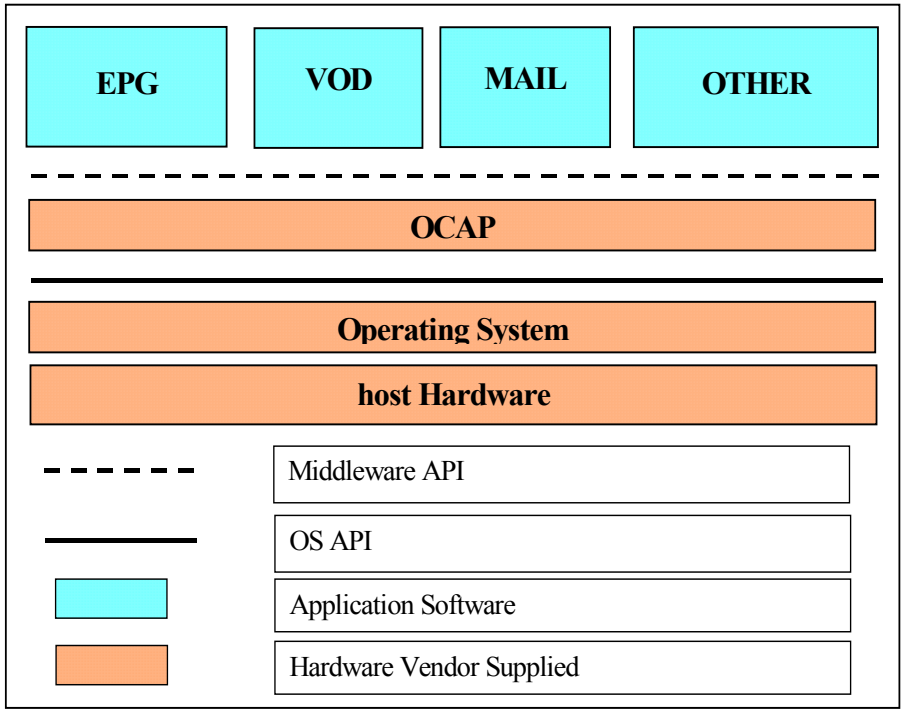
From regulation to innovation

- Born out of 1996 Telecom Act
  - FCC tasked with creating an STB competitive market
- Separable security—the CableCARD
  - Remove Conditional Access from host
  - Communication specifics decoupled
- Create a retail environment
- What about application portability?…

Enter OCAP

OpenCable Application Platform

- OCAP is just **middleware!**
  - Abstracts host and network specifics to defined APIs



Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

OCAP—Extensions

OCAP Development Example

The Alphabet Soup

"The nice thing about standards…"

- Many OCAP APIs expose functionality defined elsewhere:
  - The MPEG2 spec: ISO/IEC 13818
  - OpenCable
  - SCTE
  - CEA-EIA
  - IETF

DVB-MHP/DVB-GEM

DVB Defined MHP

↓

GEM Is A Subset of MHP

↓

OCAP 1.0 Is Based on GEM

(So Is Java Technology-Based Blu-ray Disc!)

Java Platform

- GEM 1.0.2 based on pJava 1.2a
  - Java 1.1.8 with some deviations
  - PBP is permitted
- OCAP 1.1 will be based on CDC PBP
- STBs are resource constrained
- Already deployed boxes are a challenge

Java TV™ API

- The Xlet and its lifecycle
  - Create, Initialize, Start, Pause, Destroy
- Signalling not defined
- javax.tv.service—access to SI data
- javax.tv.media—extensions to Java Media Framework API

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

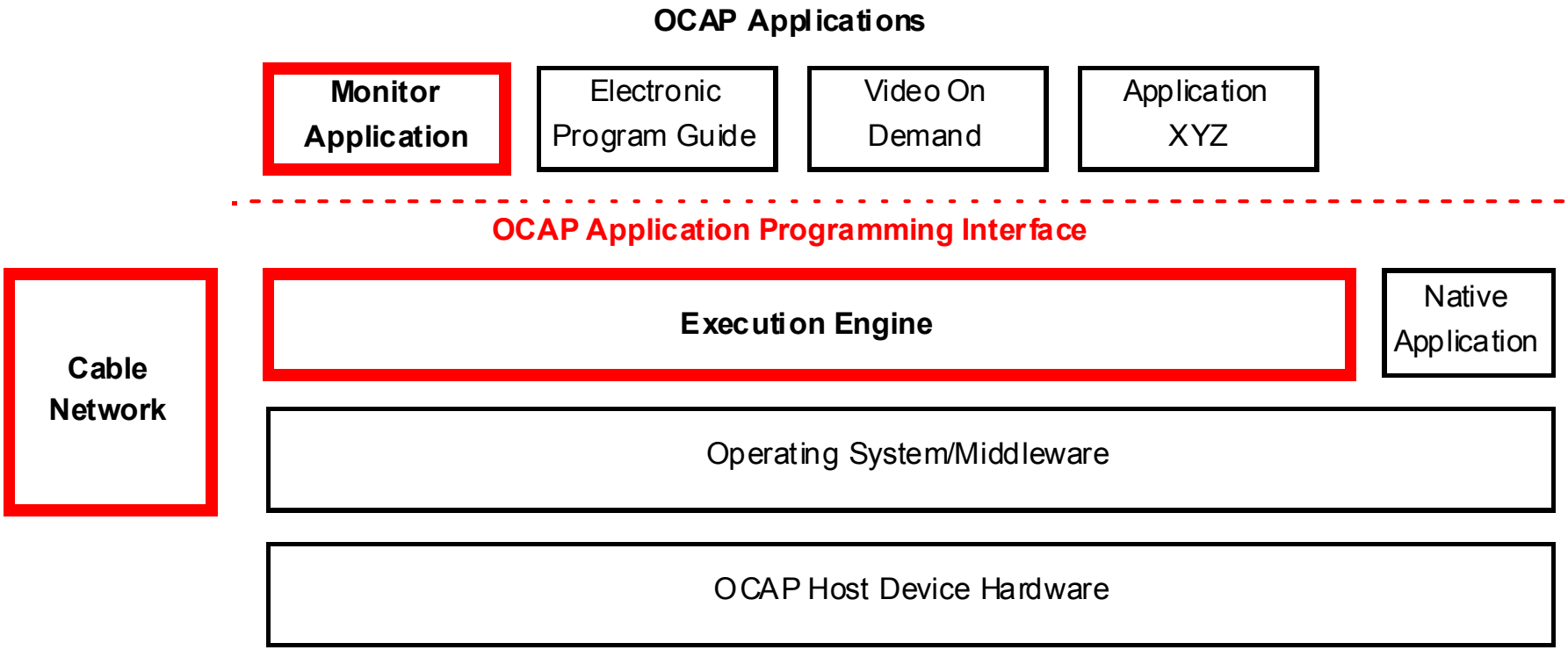
OCAP—Extensions

OCAP Development Example

Major Differences From MHP

- Unbound applications
  - The Monitor App
- Resource management
- North American SI APIs

The Basic Architecture



Application Model

- Service bound applications
  - The Application Information Table
- Unbound applications
  - The XAIT
  - Abstract services
- Service APIs
- Application priority

Priority Range	Application Type
255	Initial Monitor App
100 to 254	Unbound App
1 to 200	Service Bound App

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

OCAP—Extensions

OCAP Development Example

org.ocap.si

Provides a mechanism for accessing broadcast service information

- SI on OCAP systems deviates from that on DVB
- OCAP uses Java TV Extension SI APIs
  - javax.tv.service.*
  - javax.tv.locator
- This package provides APIs specific to OpenCable

org.ocap.si

Code example

```
public class InbandListener implements TableChangeListener
{
          Locator locator = OCAP_URL;
    private int eventCount = 0;

    public void setupListener() {
        ProgramAssociationTableManager m =
ProgramAssociationTableManager.getInstance();
        m.addInBandChangeListener(this, locator );
    }

    /* A TableChangeListener.  Counts the SIChangeEvents received. */
    public synchronized void notifyChange(SIChangeEvent event) {
        eventCount++;
        this.notify();
    }
}
```

org.ocap.service

Provides a representation of OCAP specific services

- Exposes abstract service information via AbstractService interface
  - Complementary to javax.tv.service.Service
- Provides permission and resource usage of services

org.ocap.system

Used to implement assumable system modules

- OCAP Apps with sufficient permissions can assume certain system modules
  - Private Host Application
  - MMI Resource
  - Watch TV
  - EAS
- App sends/receives APDUs to/from CableCARD

org.ocap.system

Code example

```
public class PrivateHostApp implements SystemModuleHandler{
          SystemModule sm;
    SystemModuleRegistrar registrar;

    byte[] privateHostAppID = {0x01, 0x23, 0x45, 0x67};

    public PrivateHostApp() {
          registrar = SystemModuleRegistrar.getInstance();
          // Note that a current resident Private Host Application
          // (ID=0x01234567) will terminate by this method call.
          registrar.registerSASHandler(this, privateHostAppID);
    }

    public void ready(SystemModule systemModule) {
       // Note that this is not an actual byte data.
       int    sasDataRqstApduTag = 0x9F9A02;
       byte[] sas_data_rqst_byte = {0x01, 0x23, 0x45, 0x67};

       //Start communication with the POD.
       sm = systemModule;
       sm.sendAPDU(sasDataRqstApduTag, sas_data_rqst_byte);
    }
}
```

org.ocap.application

Provides APIs for controlling app lifecycle

- Alternative method of registering unbound apps
- Overrides org.dvb.application.AppAttributes
- Provides method to filter applications

org.ocap.application

Code example

```
public class MARegistering {

    public MARegistering(byte [] xaitData) {

        AppManagerProxy appMgrProxy = AppManagerProxy.getInstance();

        /* register unbound applications. */
        try {
            ByteArrayInputStream xait =
                    new ByteArrayInputStream(xaitData);
            appMgrProxy.registerUnboundApp(xait);
        }
        catch (IOException e) {
        }

        /* unregister an unbound application that is registered by a    */
        /* sample XAIT defined by the xaitData.                         */
        AppID appid = new AppID(0x1234, 0xABCD);
        appMgrProxy.unregisterUnboundApp(appid);
    }
}
```

org.ocap.hardware

Exposes OpenCable specific hardware properties

- Copy control information about outputs
- Host, 1394, PowerModeChangeListener
- VideoOutputPort

org.ocap.hardware

Code example

```
public class DisableVideoOutput
{

          public void disableVideoOutput() {

          //Get VideoOutputPorts of the Host.
    Host host = Host.getInstance();
    Enumeration vps = host.getVideoOutputPorts();

    //For each VideoOutputPort:
    while (vps.hasMoreElements()) {
        VideoOutputPort vp = (VideoOutputPort) vps.nextElement();

        //Call disable().
        try {
            vp.disable();
        }
        catch (SecurityException e) {}
        catch (Exception e) {}
    }
}
```

org.ocap.hardware.pod

Provides access to CableCARD resource

- Application Information Resource
  - CableCARD may have applications
    - CA, Diag, IP Service, Copy Protection
- Generic feature control
  - OCAP apps can modify host device features via CableCARD

```
import org.ocap.hardware.pod.*;
...
POD pod = POD.getInstance();
int acOutlet = 7;
byte[] unswitched = {0X02};
byte[] value = pod.getHostParam(acOutlet);
...
pod.updateHostParam(acOutlet, unswitchted);
...
```

org.ocap.media

Provides OCAP specific implementations and extensions to javax.media

- Closed captioning within a Java Media Framework API player
- Allows an OCAP app to control access to media
- Provides VBI filtering mechanism

org.ocap.net

Provides a representation of an OCAP URL

- Locators provide pointers to content
  - Services
    - Different SI requires different locators
    - OOB FDC of CableCARD
  - Components
  - Files

```
OcapLocator locator = null;
 try {
     locator = new OcapLocator(TestOcapLocator.VALID_SOURCEID1);
     System.out.println(locator.toExternalForm());
 } catch (Exception e) {
     log.fail("OcapLocator(int) threw an exception.", e);
 }
```

Source: OCAP Tutorial—The Interactive TV Web http://www.interactivetvweb.org/tutorial/ocap/locators.shtml

org.ocap.event

Provides management of user input events

- User events generated by platform to signal remote key press
- Can be caught by:
  - java.awt event model: interaction is with ui
  - org.dvb.event model: interaction isn't associated with ui

```
        ...
    EventManager em = (EventManager) EventManager.getInstance();
    UserEventRepository rep = new UserEventRepository("R1");
    rep.addKey(KEYCODES[0]);
    em.setFilteredRepository(rep);
...
```

org.ocap.system.event

Defines events an app can register to receive

- ErrorEvent: an uncaught exception or implementation error
- RebootEvent: exposes how reboot was initiated
- ResourceDepletionEvent:
  - CPU Bandwidth Depleted
  - RC Bandwidth Depleted
  - System or VM Memory Depleted
- SystemEvent: 26 types defined

org.ocap.system.event

Code example

```
public class EventListenerAppSample implements SystemEventListener
{
    private final static int MAX_EVENT_STORE = 5;
    private static int eventCount = 0;
    private SystemEvent[] imeStore = new SystemEvent[MAX_EVENT_STORE];

    public EventListenerAppSample()
    {
        SystemEventManager sem = SystemEventManager.getInstance();

        // Set this object as the new error event listener.
        sem.setEventListener(SystemEventManager.ERROR_EVENT_LISTENER, this);
    }

    public void notifyEvent(SystemEvent me)
    {
        imeStore[eventCount] = me;      // Store the event for later retrieval.
    }
}
```

org.ocap.resource

Allows monitor app to manage resource contention

- Monitor app can implement and register a ResourceContentionHandler to receive warnings
- Warning thresholds can be set
- Exposes app/resource correlation

org.ocap.resource

Code example

```
public class ResourceHandler extends AppsDatabaseFilter
        implements ResourceContentionHandler {

    private static final int REJECTED_ORGANIZATION = 0xABCD;
    private static final int PRIORITIZED_ORGANIZATION = 0x1234;

    public ResourceHandler() {
        super();
        ResourceContentionManager rcManager = ResourceContentionManager.getInstance();
        rcManager.setResourceFilter(this,
"org.davic.mpeg.sections.SectionFilterGroup");
        rcManager.setResourceContentionHandler(this);
    }

    public boolean accept(AppID appid) {
        if(appid.getOID() == REJECTED_ORGANIZATION) {
            return(false);
        }
        return(true);
    }
```

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

OCAP—Extensions

OCAP Development Example

DVR

- Provides APIs to allow OCAP applications to perform DVR functionality on DVR compliant hosts
- APIs are purposely abstract
  - PVR
  - Timeshifting
  - Pushed content

Home Networking

- Provides APIs and protocols to allow OCAP applications to discover personal content on a home network
- Based on UPnP standards
- Effort underway to allow cable to provide content to a home network

Agenda

Background

The Foundation

OCAP—Architecture

OCAP—APIs

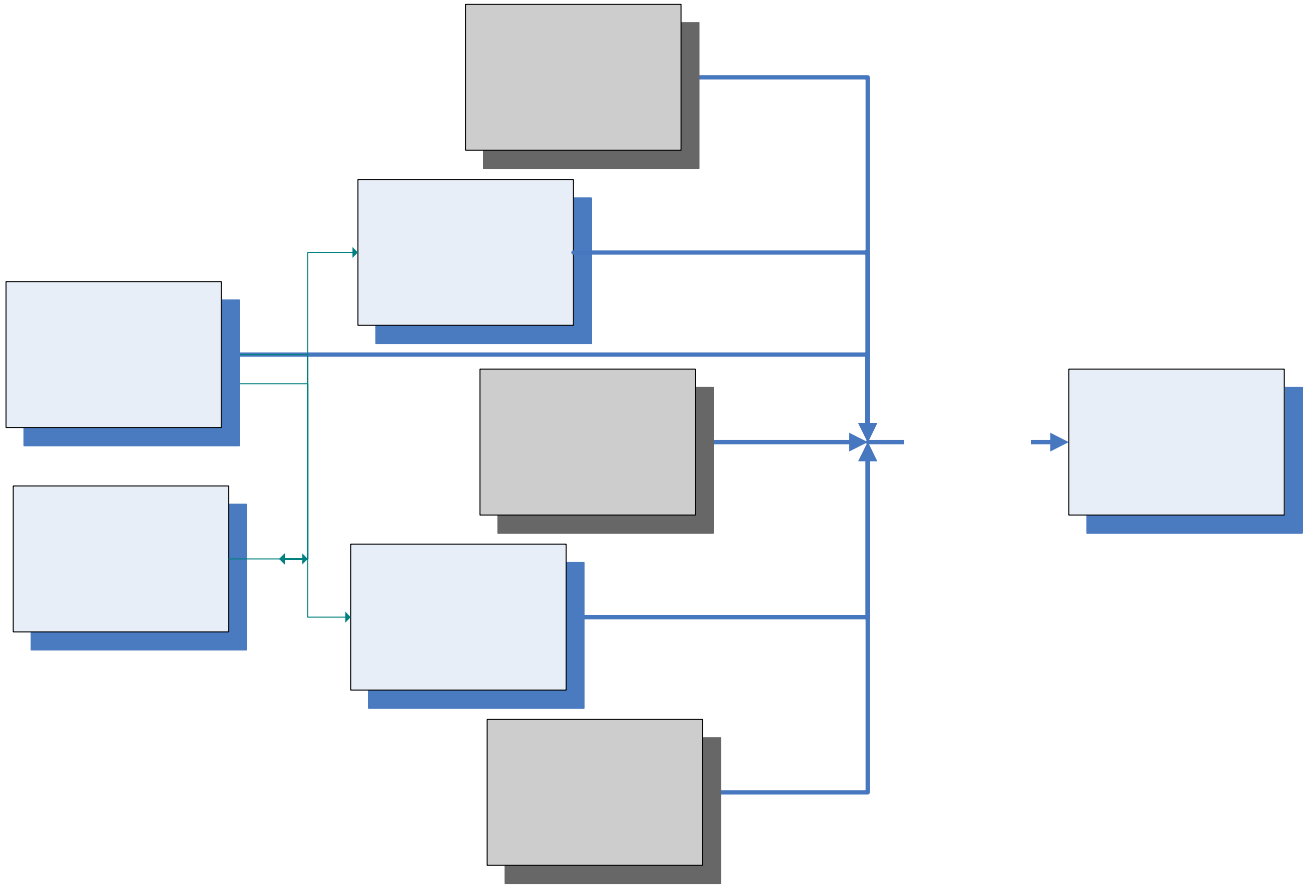
OCAP—Extensions

OCAP Development Example

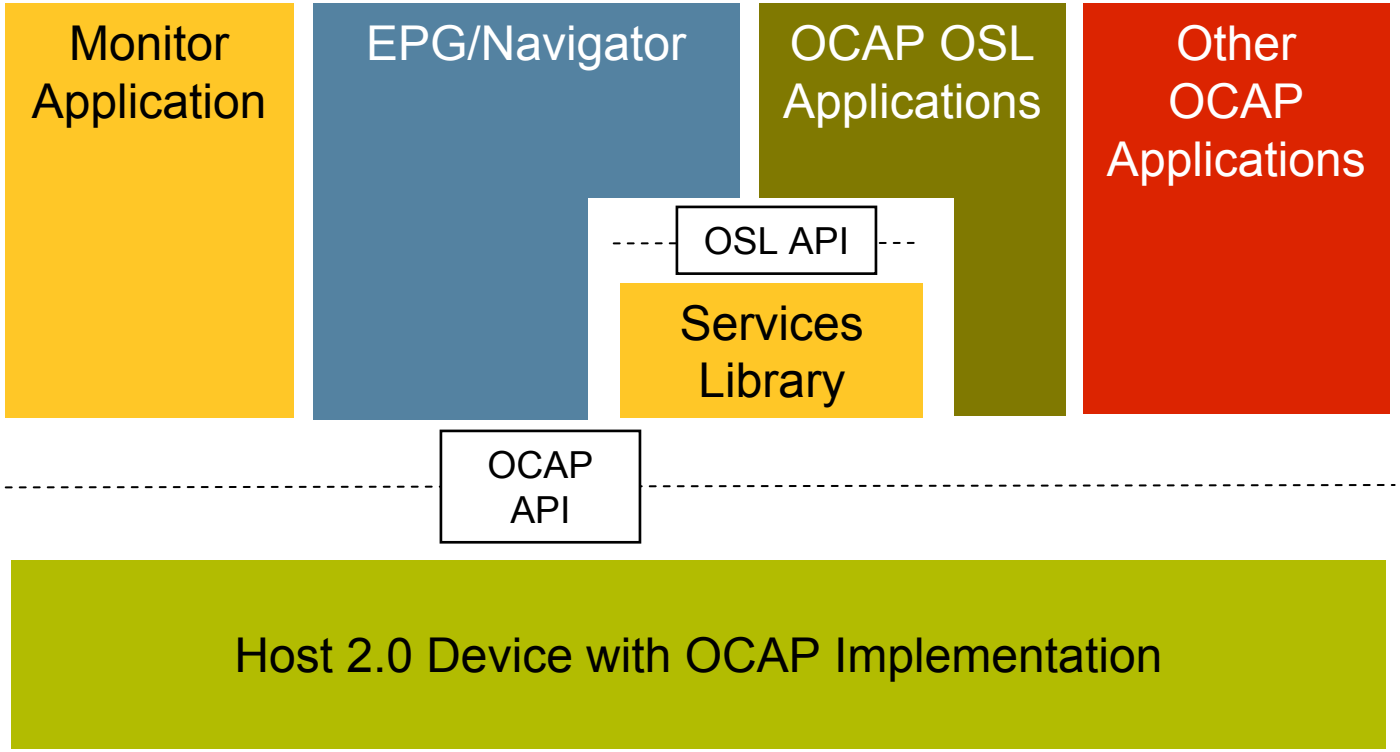
Migration to OCAP from Native

- Benefits
  - Separation of set-top box device functionality from application functionality
  - Open platform for application development
  - Reduce the barrier-to-entry for application developers
- Challenges
  - Maintain and improve the user-experience on leased devices
  - Maintain the ability to control and manage the device and its resources
  - Ensure network security, device security, and content security over an open-platform
  - Support new services without compromising any of the existing services
  - Supporting multiple OCAP stack implementations and multiple devices
  - Supporting various cable business requirements

OCAP: Comcast System Architecture



OCAP Devices: Comcast Client Architecture



Usage of OCAP Features: Monitor Application

- Monitor application enforces various MSO policies
- It uses OCAP registered library API to register Services Library modules
- It starts various server instances of the services library
- It uses OCAP Application Manager API to prevent unauthorized applications from running
- It registers a resource contention handler to manage prioritization for resource conflicts such as tuner conflicts, HScreen device reservations, and Key reservations
- Monitor system events using the System Event Manager API and take actions such as restarting an application
- Use StorageManager API and Host API to disable/enable features such as DVR or HDMI

Usage of OCAP Features: Services Library

- Java Media Framework API to implement VOD functionality
- Network Interface functionality to access tuners for direct tunes
- Carousel API and http API for file-download service
- Generic feature APIs to access PINs and PIN reset events
- Section Filtering API to access Enhanced TV Streams
- Light-weight trigger API to store Enhanced TV Streams
- SAS tunnel API to communicate with the CableCard for IPPV and CA authorization status
- MediaAccessHandler API to enforce Parental Control settings
- RecordingManager to schedule and manage recordings
- RecordingManager and ResouceContentionHandler APIs to manage conflict handling for recorded programs
- TCP/IP API for search queries and Event logging

Usage of OCAP Features: Guide Application

- Provides “Watch TV” functionality using Java TV API Service listing and Java TV API Service selection
- Exclusively reserves a set of keys for providing a consistent mapping between certain keys and functionality; for example:
  - Guide key for the guide and the Menu key for the menu-based navigation of services
  - Trick-play keys for DVR functionality
- OCAP Device Settings extension for HD settings, stereo settings, and volume control
- Video Scaling API for scaled video
- Preference API for Default language setting
- Havi API for selection of graphics configuration
- EAS Manager to be notified about System EAS handling
- Personal Java APIs for graphics, focus handling, key handling, and VM functionality

Usage of OCAP Features: DVR Application

- Uses RecordingManager to list recorded programs
- Uses Java TV API Service Selection to selection recorded programs and Java Media Framework API for controlling trick-play
- Uses TimeShift properties API and BufferingRequest API for providing foreground and background time-shift buffering and swap functionality

Usage of OCAP Features: Other Applications

- HSound API for menu sounds
- OCAP HN extension for Personal Media Management App to access Music, Photos, and Home video in UPnP devices connected to the home network
- HTTP and 2-way communications API for ETV user-agent application

Q&A

Kevin Flanagan—CableLabs

Labeeb Ismail—Comcast Cable

OCAP: A Summary of Technical Features and APIs

Kevin Flanagan

Software Engineer
CableLabs
http://www.opencable.com

Labeeb Ismail

VP of Technology
Comcast Cable
http://www.comcast.com

TS-0011