はじめよう！
Blu-ray Disc Java プログラミング

ヴィジョネア株式会社
シニアエンジニア 西村平八郎
nishimura@visionare.co.jp

Agenda

- 挨拶
- 自己紹介
- Blu-ray Disc の基礎知識
- デモ
- BD-J API 概要
- BD-J 開発の実践

自己紹介

西村平八郎

- **所属**

  ヴィジョネア株式会社
  技術開発部 シニアエンジニア

- **仕事**

  BD-J 開発＆オーサリング、サーバサイドアプリケーション開発
  ネットワークインフラ構築・管理　etc.

- **メールアドレス**

  nishimura@visionare.co.jp
  heihachirou@gmail.com

- **趣味**

  楽器演奏（ギター、ドラム、キーボード）主にブルースやブラジル音楽。

Blu-ray Disc の基礎知識

特徴

- **高画質、高音質**
  Full HD(1920x1080), H.264, VC-1, Mpeg2, 映像ビットレート Max40Mbps
  LPCM, Dolby Digital(AC3), DD+, DTS, DTS-HD etc. 7.1ch
- **大容量**
  一層 25GB, 二層 50GB, 三層・四層の仕様も (BDXL) 。
- **２系統の映像・音声同時再生**
  Primary Video/Audio, Secondary Video/Audio, Picture in Picture(PiP)
- **再生中のメニュー表示 (PopUp Menu)**
- **ローカルストレージ**
  Application Data Area, Binding Unit Data Area
- **ネットワーク対応**
  プログレッシブダウンロードによる再生にも対応。
- **コンテンツのアップデートが可能 (VFS Update)**

Blu-ray Disc の基礎知識

Player Profile について

- **Profile 1.0**
  最初に市場に出たプレイヤー、 BD-J に対応。

- **Profile 1.1(Bonus View)**
  Picture in Picture(PiP) 再生に対応。 Min. 256MB のローカルストレージを搭載。

- **Profile 2.0(BD-LIVE)**
  ネットワーク機能、プログレッシブダウンロードに対応。
  Min.1GB のローカルストレージを搭載 (*) 。

- **Profile 5.0(Blu-ray 3D)**
  3D 再生に対応。

DEMO

- 劇場版名探偵コナン天空の難破船（ロスト・シップ）

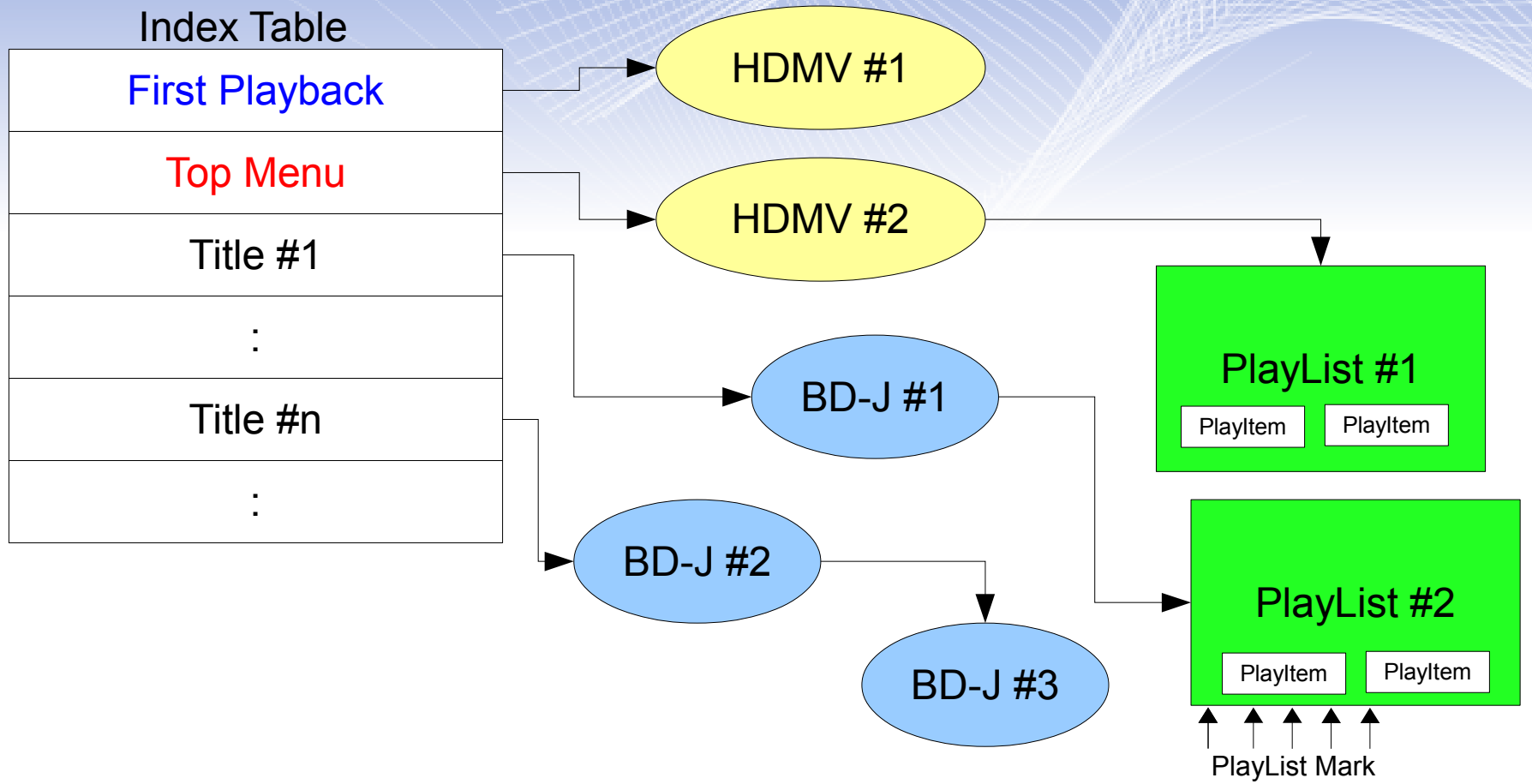
第 3 回 DEG Japan アワード
ブルーレイ大賞

ベスト・インタラクティビティ賞

ノミネート

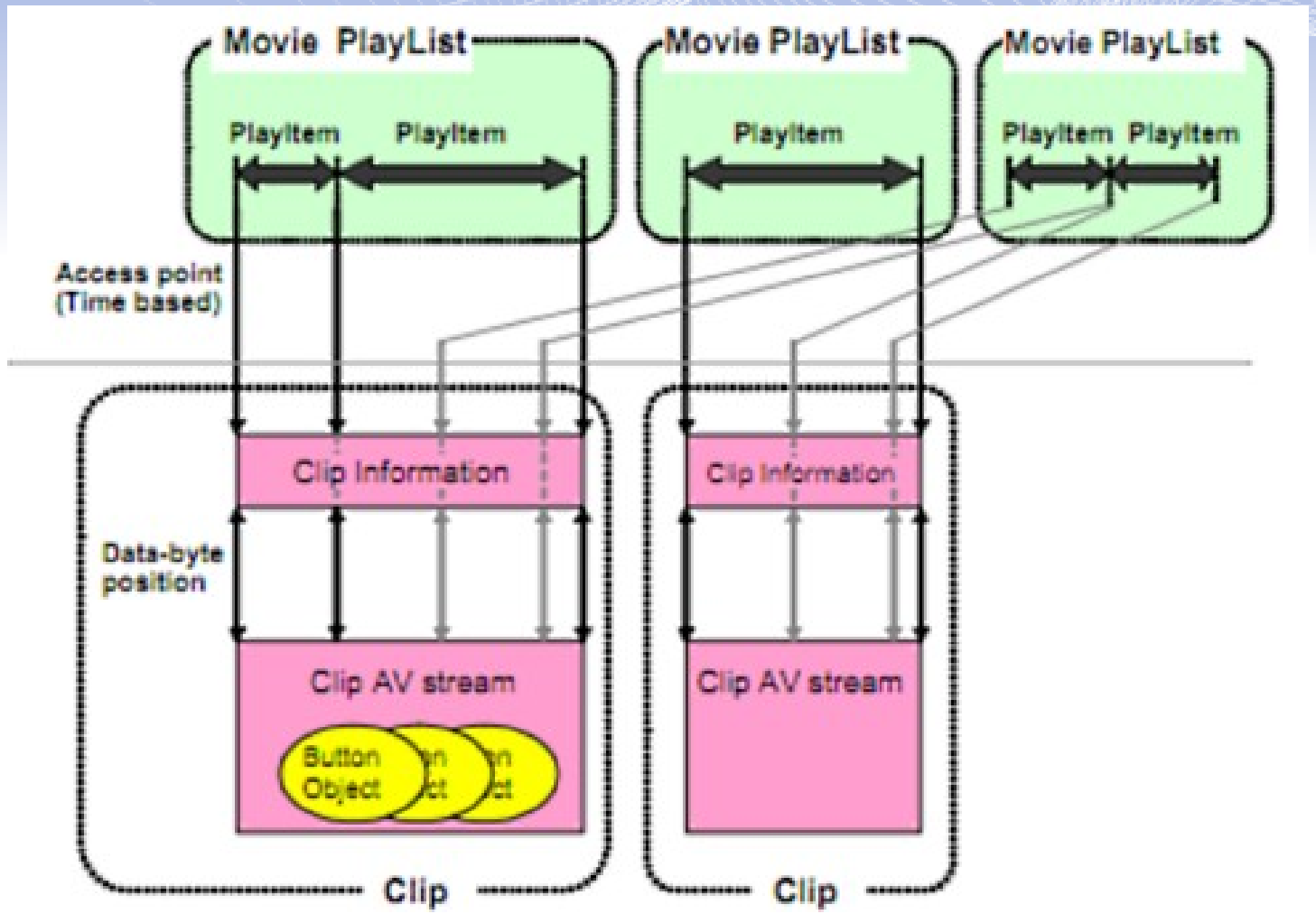
Blu-ray Disc の基礎知識

- アプリケーション構成



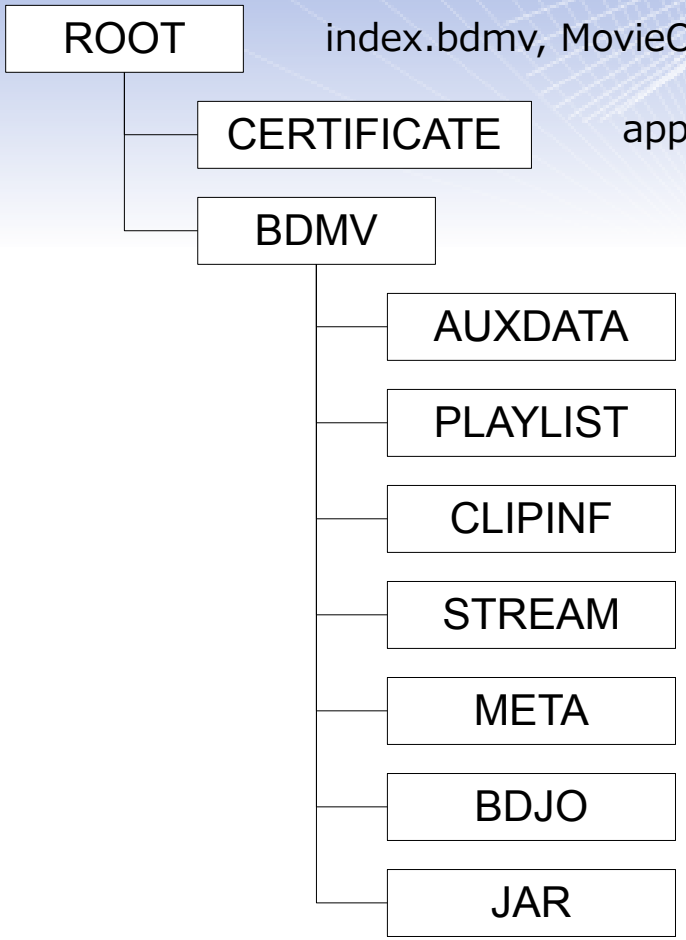
Blu-ray Disc の基礎知識

- プレイリストとクリップ



Blu-ray Disc の基礎知識

ディレクトリ構成



index.bdmv, MovieObject.bdmv

app.discroot.crt, bu.discroot.crt, id.bdmv 等。

サウンドファイル (sound.bdmv) 、フォントファイル等。

プレイリストファイル (00000.mpls) 。

クリップ情報ファイル (00000.clpi) 。

クリップ AV ファイル (00000.m2ts) 。

メタ情報。

BDJO ファイル (00000.bdjo) 。

JAR ファイル (00000.jar) 。

Blu-ray Disc の基礎知識

ファイルの説明 1

index.bdmv

Index Table(タイトル構成) を格納。

MovieObject.bdmv

HDMV Title のコマンドセットを格納。

BDMV/BDJO

zzzzz.bdjo

BD-J タイトルの各種設定・属性情報を格納。※ zzzzz は任意の半角数字

BDMV/JAR

zzzzz.jar

Xlet を構成する Java クラスファイルやリソースを格納した Jar ファイル。

BDMV/PLAYLIST

zzzzz.mpls

Playlist 。再生する PlayItem と PlayListMark の情報を格納。

Blu-ray Disc の基礎知識

ファイルの説明 2

BDMV/CLIPINF

zzzzz.clpi

Clip Infomation 。映像や音声のランダムアクセス情報を格納。

※同じファイル名 (zzzzz の部分) の m2ts ファイルと対になる。

BDMV/STREAM

zzzzz.m2ts

Clip AV Stream 。映像や音声データの MPEG-2 Transport Stream を格納。

このファイルと組になる clpi ファイルを合わせて Clip と呼称する。

Blu-ray Disc の基礎知識

ファイルの説明 3

BDMV/AUXDATA/

sound.bdmv

Interactive Audio 用の音声データを格納。音声データのフォーマットは LPCM, 48KHz, 16bit, mono/stereo 。ファイルサイズ最大 2MB 。

dvb.fontindex

フォントファイルとフォント名のマッピングを XML で記述。

zzzzz.otf

フォントファイル。フォーマットは OpenType Font 。

Blu-ray Disc の基礎知識

ファイルの説明 4

CERTIFICATE/

app.discroot.crt

JAR ファイル署名検証用証明書ファイル。

bu.discroot.crt

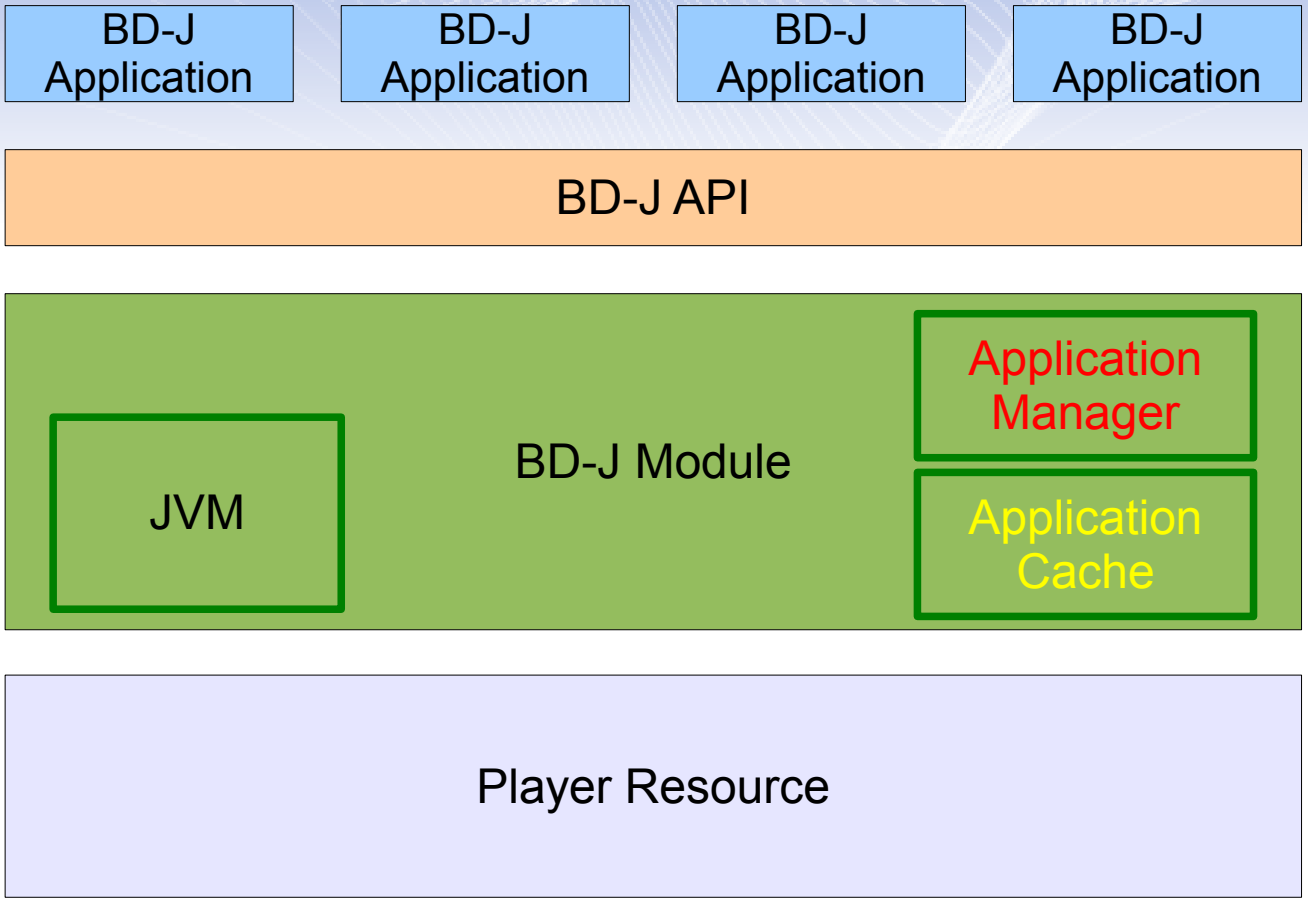
BUMF 署名検証用証明書ファイル。

id.bdmv

disc_id と **organization_id** 、 version を格納。

BD-J の基礎知識

- BD-J アーキテクチャ



BD-J の基礎知識

BD-J を構成する API 群

- **Personal Basis Profile 1.0(JSR-129)**

  http://jcp.org/aboutJava/communityprocess/final/jsr129/index.html

- **Java Secure Socket Extension (JSSE) 1.0.3**

  http://java.sun.com/products/archive/jsse/

- **Java TV 1.1(JSR-927)**

  http://jcp.org/aboutJava/communityprocess/mrel/jsr927/index.html

- **DVB-GEM 1.0.2 (Globally Executable MHP)**

  http://www.mhp.org/specs/ts_102819v010301p.pdf

- **BD-ROM Audio Visual Basic Specification Part3-2 (BD-J Specification)**

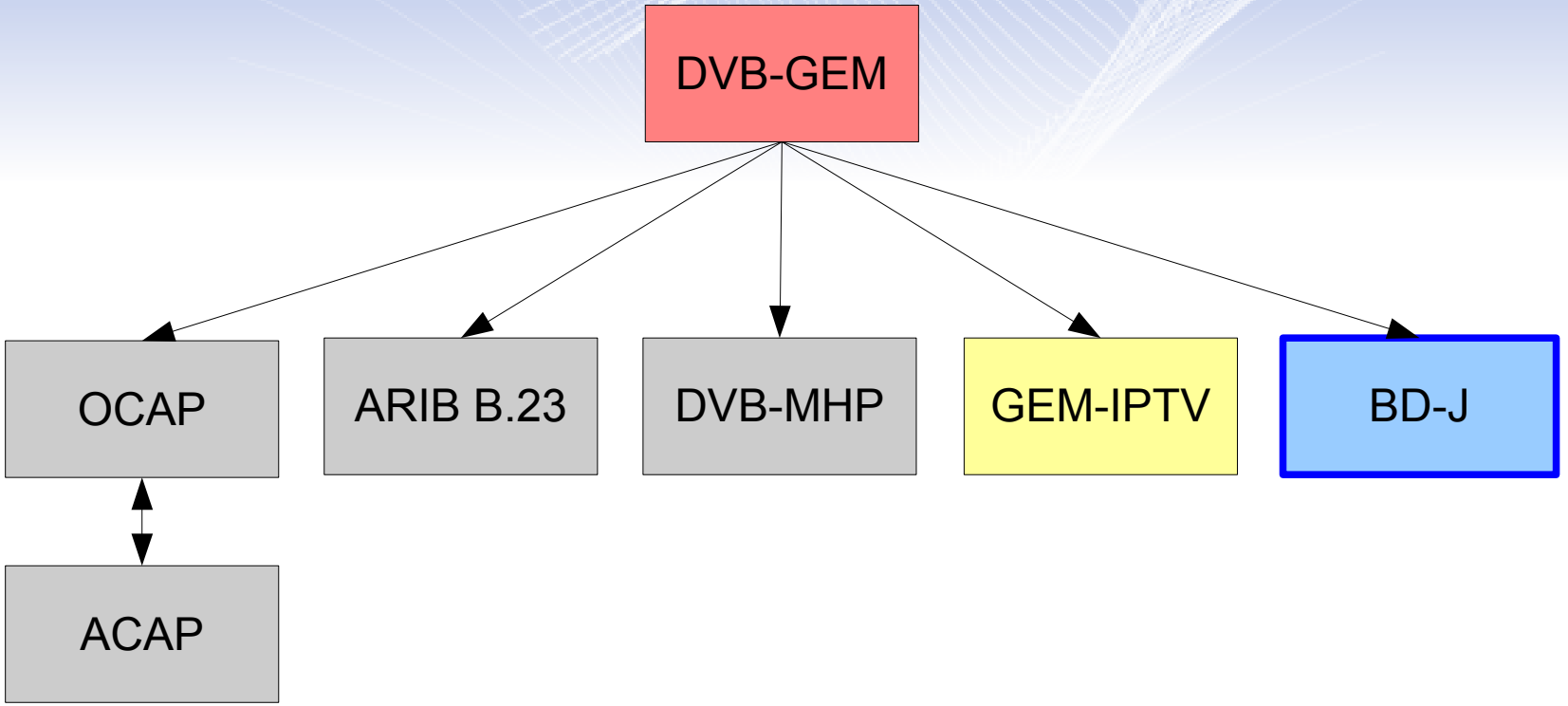
  http://www.blu-raydisc.info/

BD-J の基礎知識

- What's GEM?
  - Globally Executable MHP の略称。
  - Digital Video Broadcasting(DVB) が策定したデジタル放送向けの Java 仕様である Multimedia Home Platform(MHP) のうち、プラットフォームに依存した部分を除いて、共用できる機能を抽出したもの。
  - 各地域のデジタル放送やケーブルテレビ、衛星放送、 IP 放送などのプラットフォーム仕様の土台として採用されている。

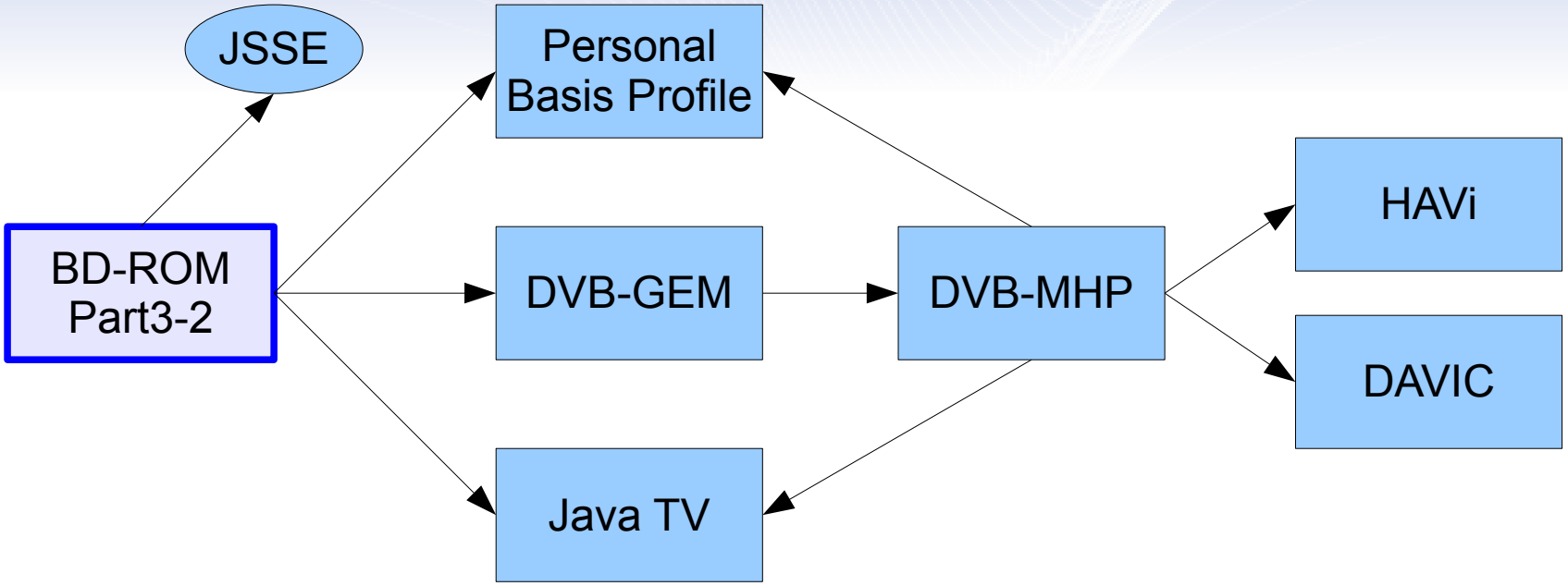
BD-J の基礎知識

- GEM Applications



BD-J の基礎知識

- API 相関図

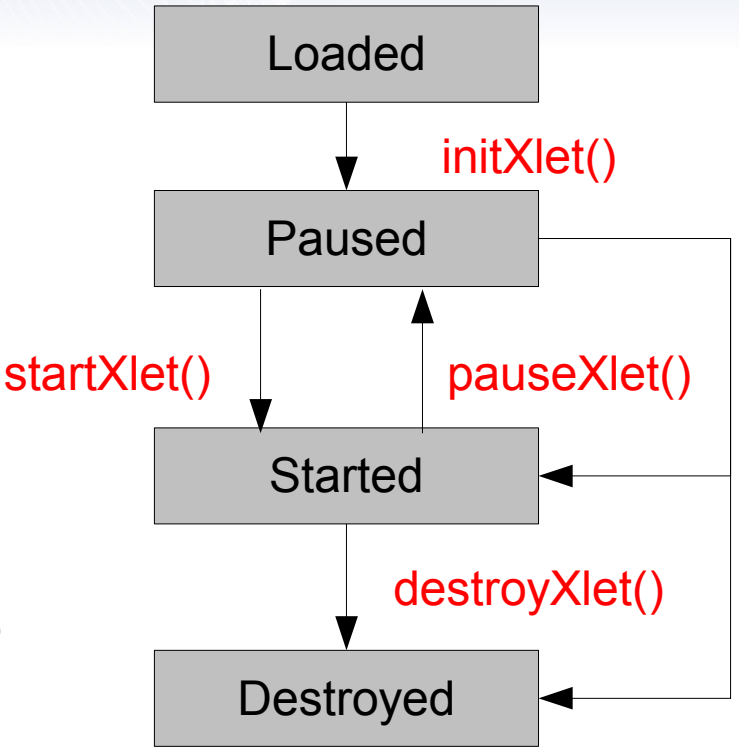


BD-J の基礎知識

Xlet について

- BD-J アプリケーションは必ず javax.tv.xlet.Xlet インターフェイスを実装したクラスをエントリポイントとする。

```
import javax.tv.xlet.XletContext;
import javax.tv.xlet.XletStateChangeException;
import javax.tv.xlet.Xlet;

public class MainXlet implements Xlet {

    public void initXlet(XletContext context)
     throws XletStateChangeException{
    }

    public void startXlet()
     throws XletStateChangeException{
    }

    public void pauseXlet(){
    }

    public void destroyXlet(boolean unconditional)
     throws XletStateChangeException{
    }
}
```



BD-J の基礎知識

BDJO(BD-J Object) について

- Application Manager が BDJO ファイルの情報を参照して BD-J タイトルのライフサイクルを制御する。
- 下記のような情報が格納される。
  - エントリポイントとなるクラス (Xlet) のパス
  - Application Cache へのクラスロード設定
  - アクセスする PlayList の指定
  - タイトル再生時の Xlet の自動実行の有無
  - イベントを受けつける再生制御キー
  - Xlet のバインディング情報
  - アプリケーション ID

BD-J の基礎知識

- Xlet のバインディング
  - Xlet の終了のタイミングによって下記の 3 種に分類できる。
    - **Title Bound**
      別のタイトルへの移動すると Xlet 終了。
    - **Title Unbound Disc Bound**
      再生停止、またはディスクをイジェクトすると Xlet 終了。
    - **Disc Unbound**
      ディスクをイジェクトしても Xlet は終了せず、次に
      挿入したディスクの First Playback Title で同じ Xlet が
      指定されている場合、その Xlet の実行を継続する。

BD-J の基礎知識

各種 ID について

- **ORGANIZATION ID**

  コンテンツの所有者を識別する 32bit unsigned int の ID 。

  0x7fff0000 ～ 0x7fffffff の範囲を強く推奨。

- **DISC ID**

  ディスクを識別する任意の 128bit unsigned int の ID 。

- **APPLICATION ID**

  Xlet を識別する 16bit unsigned int の ID 。

  署名なし Xlet => 0x0000 ～ 0x3fff

  署名あり Xlet => 0x4000 ～ 0x7fff

BD-J API 概要

グラフィックス

- 「バックグラウンド」「ビデオ」「グラフィックス」の三層プレーン構成。

- 大まかに言って Java 1.3 時代の AWT から Widget を省いたもの。

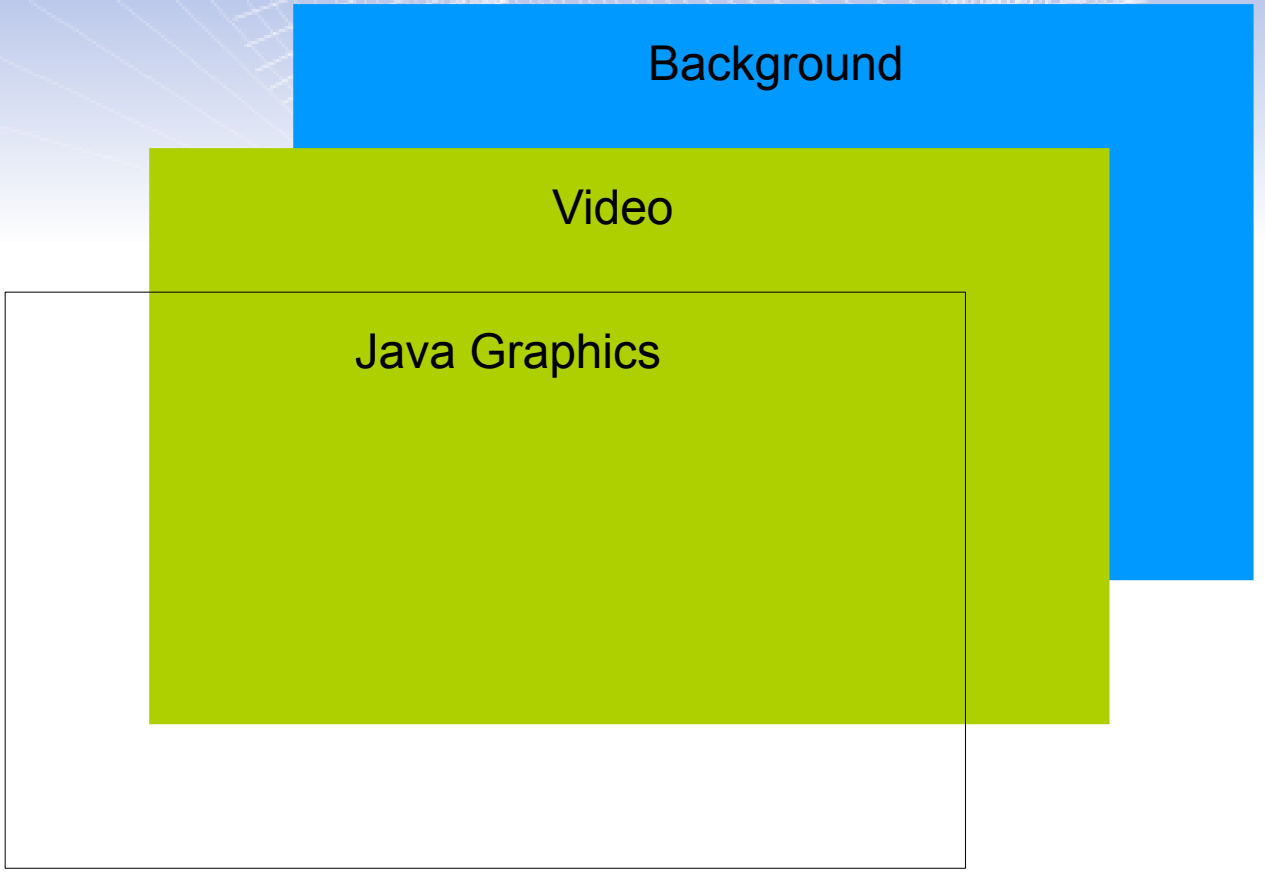
- 画像形式は JPEG, PNG, GIF 。 PNG の場合は透過表現も可能。

- 同時に再生するビデオの解像度によってグラフィックスプレーンの解像度が制限される。

  - ビデオ 1920x1080 ⇒ 1920x1080(HD), 960x540(QHD)
  - ビデオ 1280x720 ⇒ 1280x720
  - ビデオ 720x480 ⇒ 720x780(SD)

BD-J の基礎知識

- プレーン構成



BD-J API 概要

ユーザーインターフェイス

- EventManager を用いて Component にフォーカスの当たっていない場合にキーイベントを取得可能。(PopUp メニューボタン等に有効)

```
import org.dvb.event.EventManager;
import org.dvb.event.UserEvent;
import org.dvb.event.UserEventListener;
import org.dvb.event.UserEventRepository;
import org.bluray.ui.event.HRcEvent;

UserEventRepository userEventRepo = new UserEventRepository("x");
userEventRepo.addAllArrowKeys();
userEventRepo.addAllColourKeys();
userEventRepo.addAllNumericKeys();
userEventRepo.addKey(HRcEvent.VK_ENTER);
userEventRepo.addKey(HRcEvent.VK_POPUP_MENU);

EventManager.getInstance().addUserEventListener(listner, userEventRepo);
```

```
public void userEventReceived(UserEvent e) {
    int type = e.getType();
    if (type == HRcEvent.KEY_PRESSED) {
        int code = e.getCode();
    }
```

BD-J API 概要

- 任意のフォントファイル (OpenType) を用いてのテキスト描画

AUXDATA/00000.otf

AUXDATA/dvb.fontindex

```
<?xml version="1.0"?>
<!DOCTYPE fontdirectory PUBLIC "-//DVB//DTD Font Directory 1.0//EN"
"http://www.dvb.org/mhp/dtd/fontdirectory-1-0.dtd">
<fontdirectory>
<font>
   <name>IPA</name>
   <fontformat>OTF</fontformat>
   <filename>00000.otf</filename>
</font>
</fontdirectory>
```

※dvb.fontindex という XML ファイルで
フォントファイルに対応するフォント名を
記述する。

```
import org.dvb.ui.FontFactory;

Font font = new FontFactory().createFont("IPA", java.awt.Font.PLAIN, 48);
Container gui = new Container() {
    public void paint(Graphics g) {
        //フォントをセット
        g.setFont(font);

        //背景を描画
        g.setColor(new Color(10,10,255));
        g.fillRect(20, 20, getWidth()-40, getHeight()-40);

        //文字を描画
        g.setColor(new Color(245,245,245));
        int message_width = g.getFontMetrics().stringWidth("Hello, BD-J!");
        g.drawString(message, (getWidth() - message_width) / 2, 500);
    }
};
```

BD-J API 概要

再生制御 1 プレイリストの指定と再生

```
import javax.media.Control;
import javax.media.ControllerEvent;
import javax.media.ControllerListener;
import javax.media.Manager;
import javax.media.Player;
import javax.media.EndOfMediaEvent
import org.bluray.net.BDLocator;
import org.davic.media.MediaLocator;

BDLocator loc = new BDLocator("bd://PLAYLIST:00001");
MediaLocator ml = new MediaLocator(loc);
Player player = Manager.createPlayer(ml);
player.start();

//再生制御
player.setRate(0.0F);  //Puase
player.setRate(1.0F);  //Restart
player.setRate(2.0F);  //fast forward
player.setRate(0.5F);  //slow
player.stop();         //stop

//プレイヤーの状態を取得
player.addControllerListener(listener);


public void controllerUpdate(ControllerEvent event){
    if(event instanceof EndOfMediaEvent){
        //最後まで再生した場合
    }
}
```

BD-J API 概要

再生制御２　各種コントローラの取得

```
import javax.media.Control;
import javax.media.Player;
import javax.tv.media.AWTVideoSizeControl;
import org.bluray.media.PiPControl;
import org.bluray.media.PlayListChangeControl;
import org.bluray.media.PlaybackControl;
import org.bluray.media.PrimaryAudioControl;
import org.bluray.media.PrimaryGainControl;
import org.bluray.media.SubtitlingControl;
import org.davic.media.MediaTimeEventControl;
import org.davic.media.MediaTimePositionControl;

Control[] controls = player.getControls();
for (int i = 0; i < controls.length; i++) {
     if (controls[i] instanceof PlayListChangeControl) {
               playlistControl = (PlayListChangeControl) controls[i];
      } else if (controls[i] instanceof PlaybackControl) {
               playbackControl = (PlaybackControl) controls[i];
      } else if (controls[i] instanceof SubtitlingControl) {
               subtitlingControl = (SubtitlingControl) controls[i];
      } else if (controls[i] instanceof PrimaryAudioControl) {
               audioControl = (PrimaryAudioControl) controls[i];
      } else if (controls[i] instanceof PrimaryGainControl) {
               gainControl = (PrimaryGainControl) controls[i];
      } else if (controls[i] instanceof MediaTimePositionControl){
               timePositionControl = (MediaTimePositionControl)controls[i];
      } else if (controls[i] instanceof PiPControl){
               pipControl = (PiPControl)controls[i];
      } else if (controls[i] instanceof AWTVideoSizeControl){
               videoSizeControl = (AWTVideoSizeControl)controls[i];
      } else if (controls[i] instanceof MediaTimeEventControl){
              timeEventControl = (MediaTimeEventControl)controls[i];
      }
}
```

BD-J API 概要

再生制御 3 PlaybackControl

チャプタージャンプのサンプルコード。

チャプターポイントは” PlayListMark” として PlayList にあらかじめ定義しておく必要がある。

```
Control[] controls = player.getControls();
for (int i = 0; i < controls.length; i++) {
      if (controls[i] instanceof PlaybackControl) {
            playbackControl = (PlaybackControl) controls[i];
      }
}

int playListMarkId = 1;
playbackControl.skipToMark(playListMarkId);   //指定のチャプタにジャンプ
```

BD-J API 概要

再生制御４　サウンドファイルの再生

```
import javax.media.Control;
import javax.media.ControllerEvent;
import javax.media.ControllerListener;
import javax.media.Manager;
import javax.media.Player;
import javax.media.EndOfMediaEvent
import org.bluray.net.BDLocator;
import org.davic.media.MediaLocator;

BDLocator loc = new BDLocator("bd://SOUND:00");
MediaLocator ml = new MediaLocator(loc);
Player player = Manager.createPlayer(ml);
player.start();
```

BD-J API 概要

タイトルの制御

Top Menu Title に遷移するサンプルコード。

```
import javax.tv.service.selection.ServiceContextFactory;
import javax.tv.service.selection.ServiceContext;
import javax.tv.service.SIManager;
import javax.tv.xlet.XletContext;
import org.bluray.net.BDLocator;
import org.bluray.ti.selection.TitleContext;
import org.bluray.ti.Title;

public void jumpToTopMenuTitle(XletContext xletContext){
    try{
        ServiceContextFactory fact = ServiceContextFactory.getInstance();
        TitleContext tc = (TitleContext)fact.getServiceContext(xletContext);
        SIManager sim = SIManager.createInstance();
        BDLocator loc = new BDLocator("bd://0"); // <= TOP MENU TITLE
        Title t = (Title)sim.getService(loc);
        tc.select(t);
    }catch(Exception e){

    }
}
```

BD-J API 概要

入出力

- 2 種のローカルストレージ
  - Application Data Area(ADA) 64KB
  - Binding Unit Data Area(BUDA) 1GB(Profile2.0)/256MB(Profile1.1)
- Blu-ray ディスクメディア
- General Purpose Register(GPR)
  - 任意に読み書き可能なレジスタ , 32bit unsigned int 。
- Player Status Register(PSR)
  - プレイヤーの各種状態を保持するレジスタ , 32bit unsinged int, read only 。
- User Preference
  - プレイヤーに登録された設定情報。ユーザ名、言語、年齢制限等。
- ネットワーク
  - TCP/IP(v4)

BD-J API 概要

ファイル入出力

```
import java.io.File;
import javax.tv.xlet.XletContext;
import org.bluray.ti.DiscManager;

private final String FS = File.separator;
XletContext xletContext = ...;

//各種IDの取得
String discId     = DiscManager.getDiscManager().getCurrentDisc().getId();
String orgId      = (String)xletContext.getXletProperty("dvb.org.id");
String appId      = (String)xletContext.getXletProperty("dvb.app.id");

//ルートディレクトリの取得
String budaRoot   = System.getProperty("bluray.bindingunit.root");
String adaRoot    = System.getProperty("dvb.persistent.root");
String discRoot   = System.getProperty("bluray.vfs.root");

//ADAへのファイル出力
new File(adaRoot + FS + orgId + FS + appId + FS + "ada.dat");

//BUDAへのファイル出力
new File(budaRoot + FS + orgId + FS + discId + FS + "buda.dat");

// ディスクメディアからの読み込み
new File(discRoot + FS + "BDMV" + FS + "JAR" + FS + "zzzzz" + FS + "vfs.dat");
```

BD-J API 概要

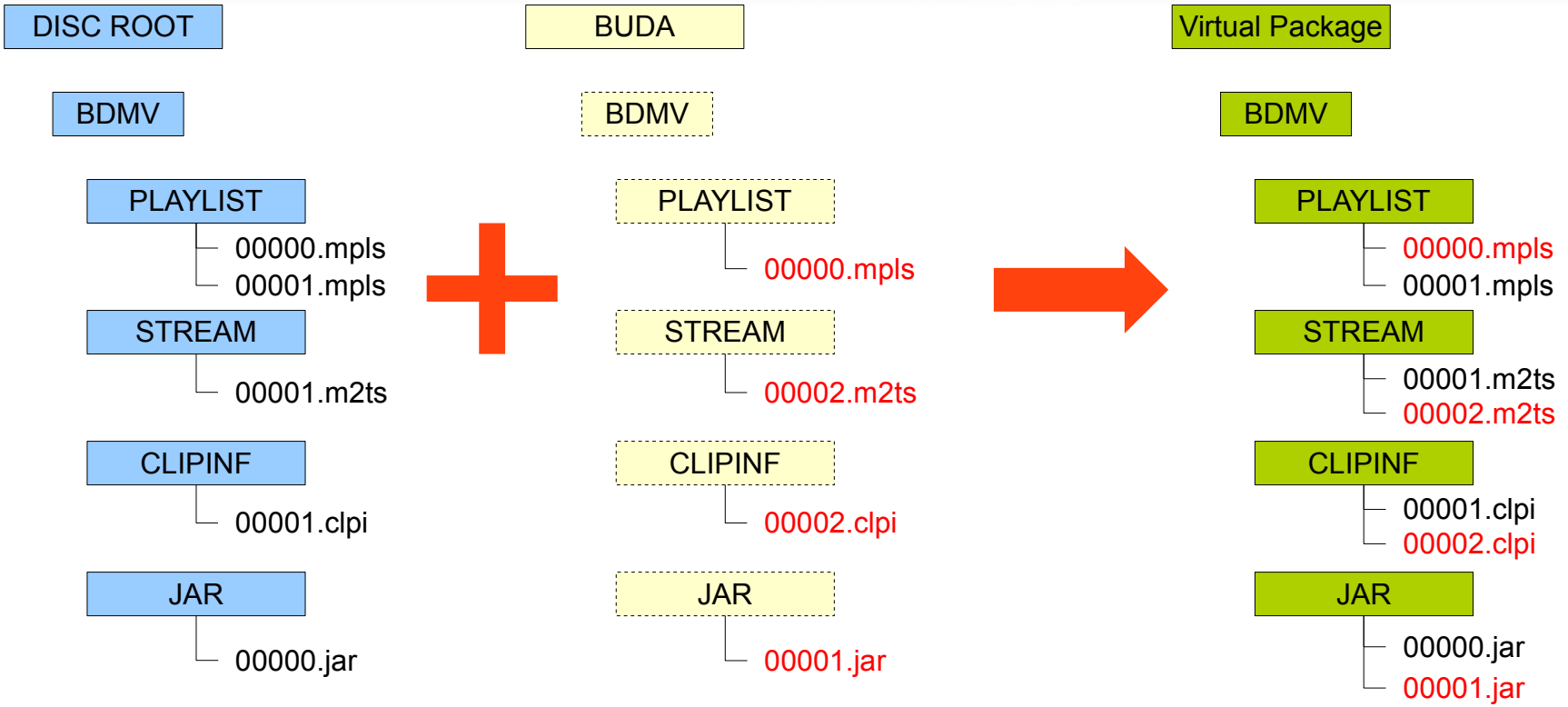
レジスタ、 User Preference

```
import org.bluray.system.RegisterAccess;

//PSR1の取得
int psr1 = RegisterAccess.getInstance().getPSR(1);

//GPR1の取得
int gpr1 = RegisterAccess.getInstance().getGPR(1);

//GPR1の設定
RegisterAccess.getInstance().setGPR(1, 12345);

//ユーザープレファレンスの取得
UserPreferenceManager manager = UserPreferenceManager.getInstance();
String[] keys = {
      "User @",
      "User Name",
      "User Language",
      "Parental Rating",
      "Country Code",
      "Default Font Size"
};

for(int i=0; i<keys.length; i++){
      GeneralPreference gp = new GeneralPreference(keys[i]);
      manager.read(gp);
      String value = gp.getMostFavourite();
      if(value != null){

      }
}
```

BD-J API 概要

VFS Update

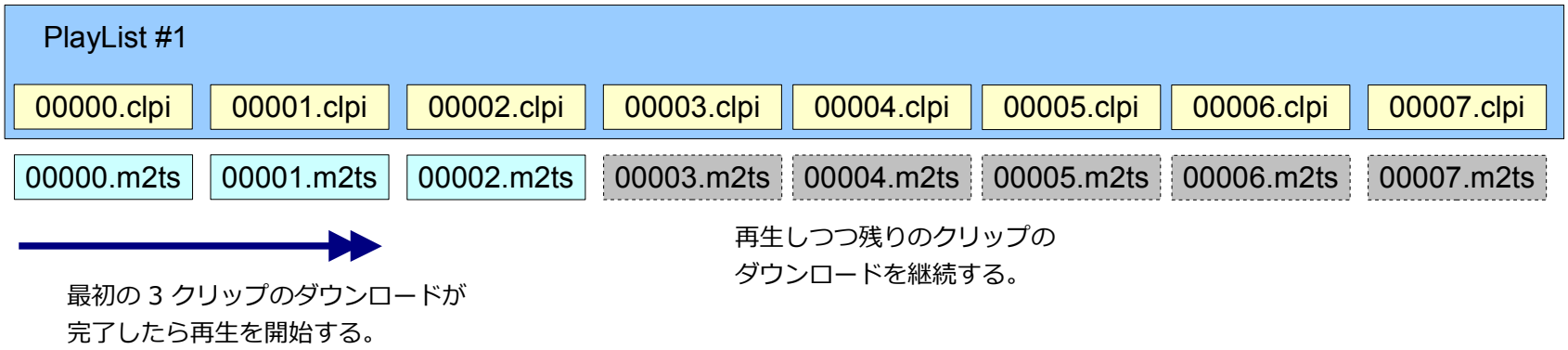
- ディスク内のファイルと BUDA 内に保存されたファイルを組み合わせて、 1 つの仮想的なディスクパッケージ (Virtual Package) に見立てて、コンテンツを更新する仕組み。
- **Binding Unit Manifest File(BUMF)** という XML ファイルで、更新するファイルのマッピングを定義する。



BD-J API 概要

Progressive Playlist

- 通常の PlayList では、再生開始時にあらかじめディスクまたは Virtual Package にストリームファイル (m2ts) が全て存在している必要があるが、 BUMF にて ProgressiveAsset として指定されているファイルについては、再生時点のクリップに対応するストリームファイルがあれば再生可能。
- これを利用すればクリップを分割してダウンロードしながら、必要な分の取得が完了したら再生を開始するというプログレッシブダウンロードが可能になり、ネットワーク越しの映像データ取得時の待ち時間を減らすことができる。



BD-J API 概要

- セキュリティ
  - コンテンツ保護

    AACS, BD+, ROM Mark
  - Online Certificate
  - アプリケーション署名

    app.discroot.crt, sha1 with RSA, BD 特有の拡張有り , Permission Request File
  - BUMF 署名

    bu.discroot.crt, sha1 with RSA

BD-J API 概要

アプリケーション署名

- 下記の機能を利用する場合はアプリケーションに署名を施し、
  かつ Permmision Request File(後述) に明示する必要がある。
  - ADA または BUDA への読み書き。
  - User Preference の取得。
  - 他の Xlet の起動。
  - タイトルの遷移。
  - VFS Update 。
  - ネットワーク機能。

BD-J API 概要

Permission Request File

-Xlet のクラスファイルと同じ階層に、 bluray.< クラス名 >.perm
というファイル名で配置する。

```
<?xml version="1.0" encoding="UTF-8"?>
<n:permissionrequestfile xmlns:n="urn:BDA:bdmv;PRF" orgid="ORG_ID" appid="APP_ID">
    <!-- ADA へのアクセス -->
    <file value="true"></file>

    <!-- 他の Xlet のコントロール -->
    <applifecyclecontrol value="true"></applifecyclecontrol>

    <!-- タイトルのコントロール -->
    <servicesel value="true"></servicesel>

    <!-- User Preference へのアクセス -->
    <userpreferences read="true" write="false"></userpreferences>

    <!-- ネットワーク機能 -->
    <network>
       <host action="connect,listen,resolve,accept">*</host>
    </network>

    <!-- BUDA へのアクセス -->
    <bd-bindingunitarea value="true"></bd-bindingunitarea>

    <!-- VFS Update -->
    <bd-vfs value="true"></bd-vfs>
</n:permissionrequestfile>
```

BD-J 開発の実践

- 開発環境の整備

  必要なもの

  - 開発機
  - JDK
  - BD-J プラットフォームの JavaDoc & スタブファイル
  - エディタ、 IDE 等
  - ライティングソフト (UDF2.5 に対応していること)
  - BD ブランクメディア (BD-R, BD-RE)
  - Blu-ray プレイヤー (ハードウェア、ソフトウェア)

  その他必要に応じて

  - Apache Ant, Maven 等のビルドツール
  - Subversion, Git 等

BD-J 開発の実践

- JavaDoc & スタブファイルの準備

  BD-J JavaDocStubs ライセンスの締結

  http://www.blu-raydisc.info/license-app/javadocstubs-app.php

  各 API の取得とスタブ生成手順

  http://java.sun.com/javame/reference/bluray-technote.html

  ※build.sh の実行には UNIX 系環境にて sh, unzip, zip, javac, javadoc が必要。

BD-J 開発の実践

HD Cookbook Project について

- BD-J(GEM) アプリケーションの開発に必要な各種ツールや、アニメーション
  フレームワーク等を開発しているオープンソースのプロジェクト。
  開発に関する有用なノウハウやサンプルコードが集められており、
  開発者向けのフォーラムやメーリングリスト等もある。

- プロジェクト URL
  http://jovial.com/hdcookbook/

- リポジトリ URL
  https://svn.java.net/svn/hdcookbook~svn/trunk

このプロジェクトで開発されたツールを用いれば
クリップ (clpi, m2ts) を除いて、 BD-J に必要な
ファイル類は全て生成可能。

BD-J 開発の実践

GRIN(GRaphical INteractivity) について

- HD Cookbook Project で開発されている BD-J(GEM) 環境に最適化された
  グラフィックフレームワーク。シーングラフの一種。
- GRIN show file と呼ばれる独自の定義ファイルでグラフィック要素を
  記述することで、凝った視覚効果やアニメーションを手軽に実現できる。
- GRIN の概要
  http://jovial.com/hdcookbook/grin.html

- Show file syntax
  http://jovial.com/hdcookbook/javadocs/grin/javame/com/hdcookbook/grin/doc-files/index.html