



Enterprise Media Beans

Thomas Lévêque

thomas.leveque@wanadoo.fr

Sommaire

- Introduction
- La concurrence
- Motivations
- Etat de l'Art
- Spécifications
- Exemples de programmation
- Conclusion & Perspectives
- Documentation consultée

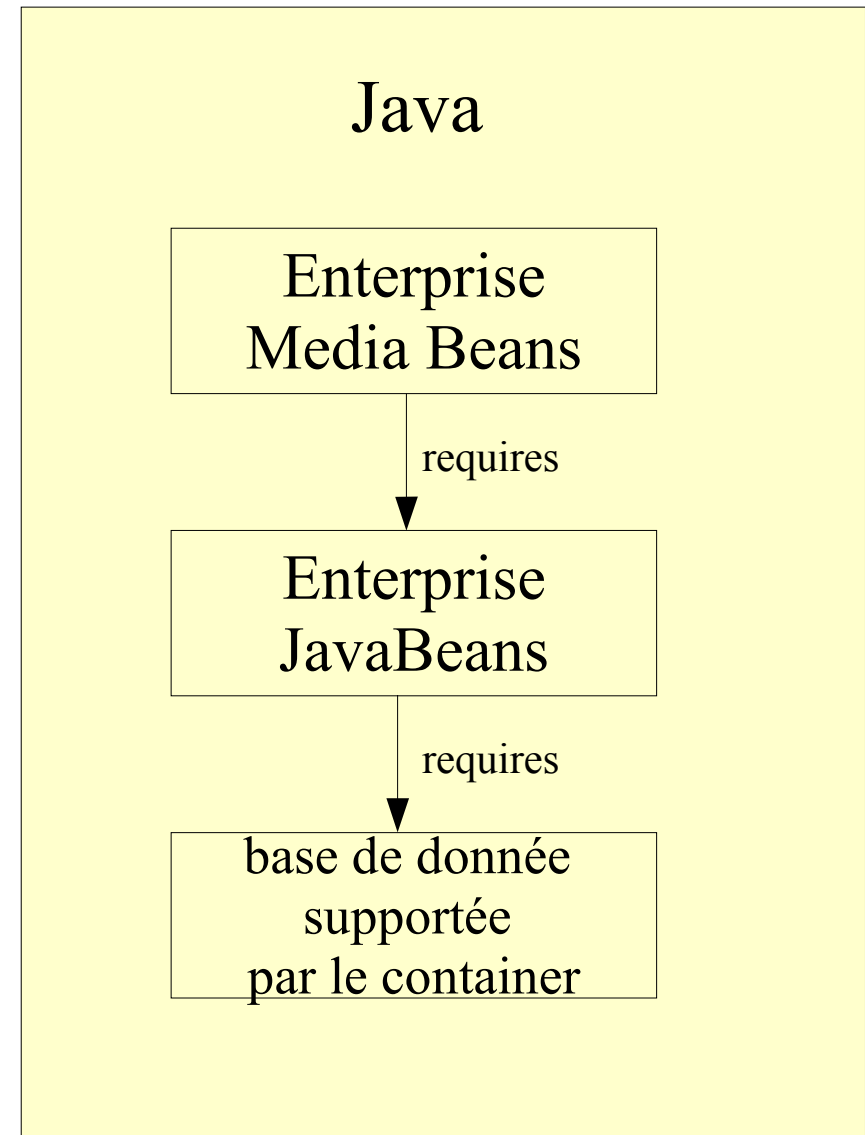
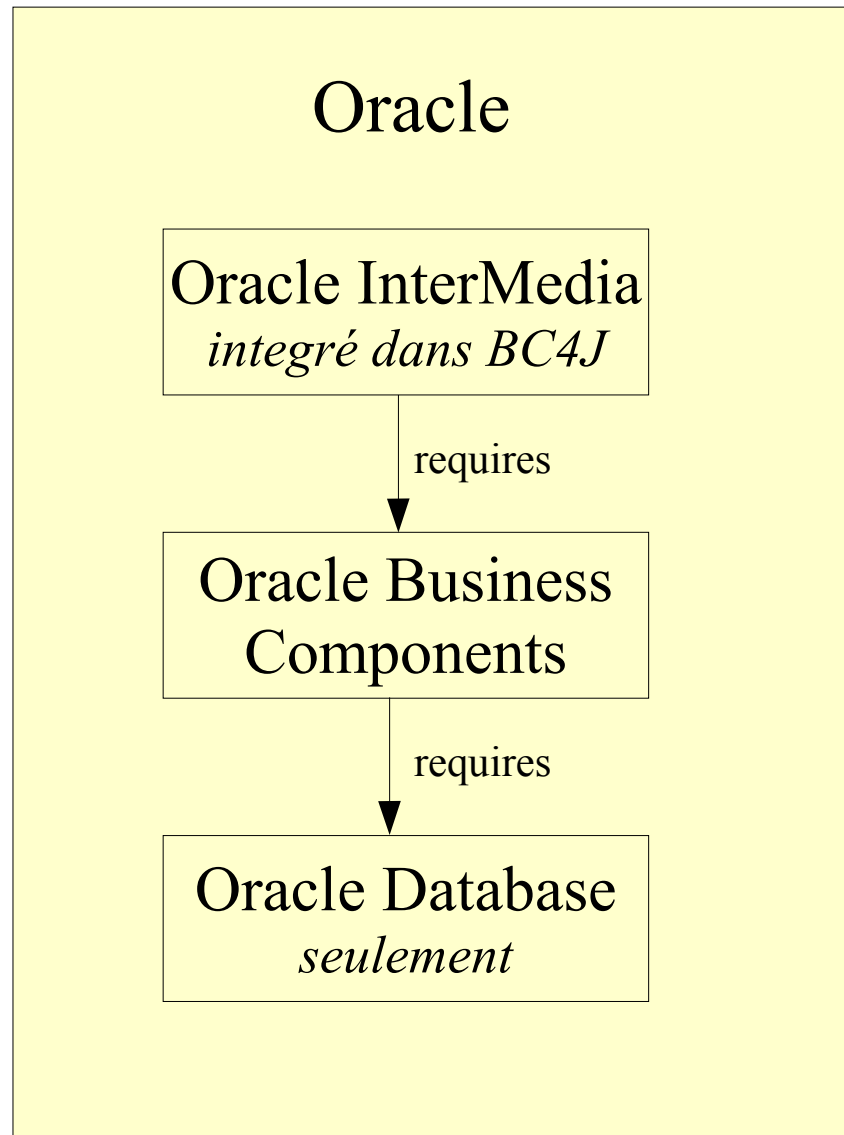
Introduction

- IBM
- JCP (Java Community Process)
- JSR 086
- Basé sur les EJB 1.1

La concurrence

- Java Media Framework
- Content Management Systems
- Media Streaming
- Oracle Intermedia

Architecture Générale



Motivations

- Définition d'une API standardisée (JSR086) pour la manipulation de médias riches
- " Ecrire une fois, Déployer n fois "
- Média riche = un type comme un autre

Etat de l'art

- Des spécifications (JSR086 v 1.0), 65 pages
- 2 packages disponibles délivrés par IBM
 - Kit de tests de validation (TCK)
 - Une implémentation de référence (RI)

Serveurs J2EE supportants les EMB

- **Supporté**
 - Websphere Application Server 4.0.2
- **Intégré**
 - Jboss 4.0

Package RI

- Contenu
 - Getting Started document
 - Source JAVA
 - Javadoc
 - Un exemple d'application Web

Package TCK

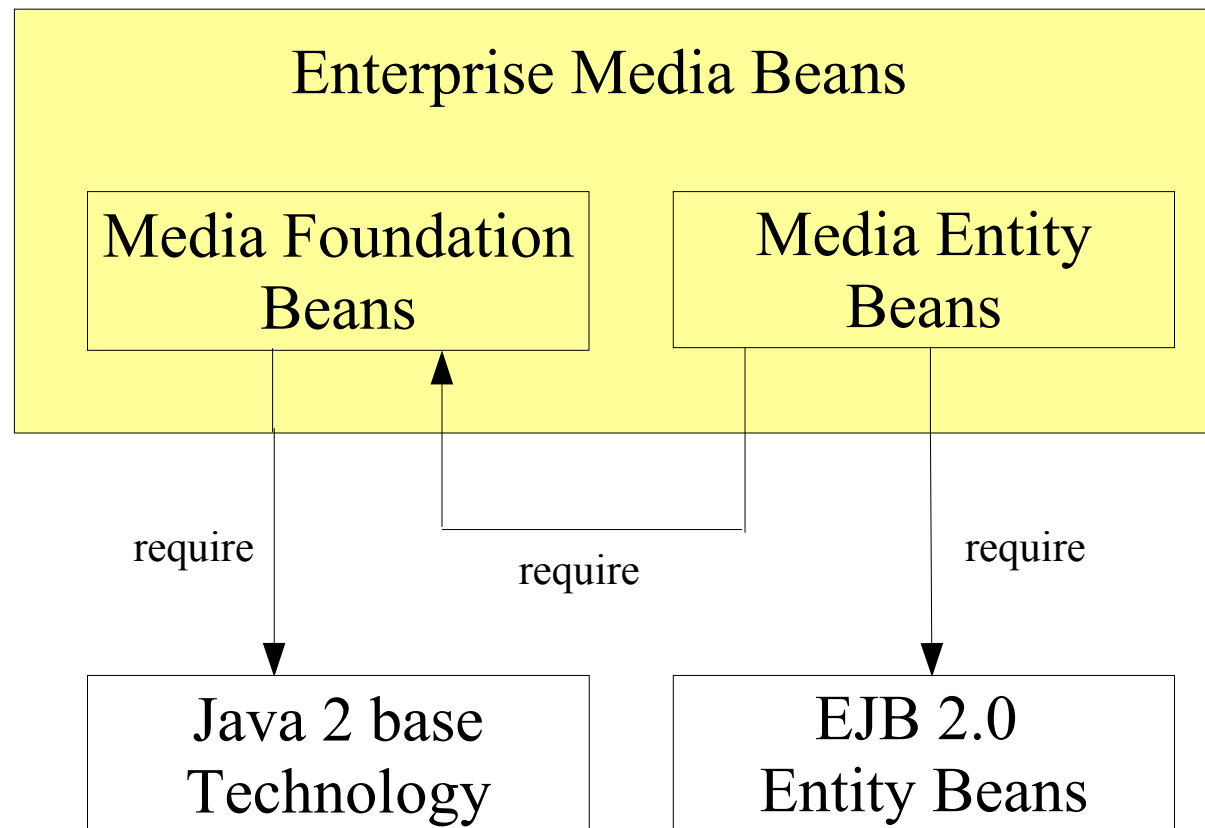
- Contenu
 - Manuel utilisateur
 - Tests JUnit
 - Javadoc
 - Session Bean Wrappers

- Packages nécessaires
 - Junit
 - Xerces

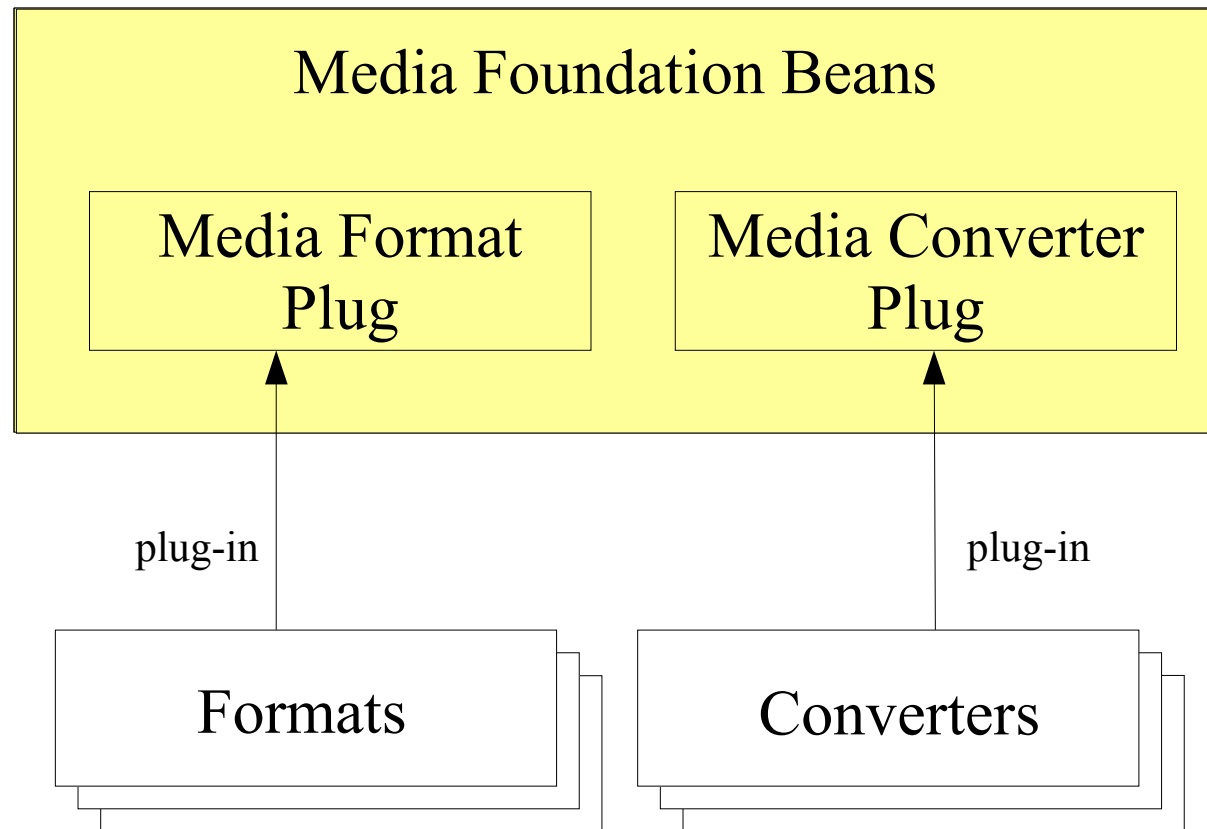
Spécifications EMB (JSR086)

- Media Entity Bean
- Media Format Bean
- Media Distribution Bean
- Exemples de programmation

Architecture des EMB



Media Foundation Beans



Media

- Interface qui étend Serializable
 - getContent()
 - getContent(long position,int length)
 - getConvertedMedia(MediaConverterSpec[] specifications)
 - getFormat()
 - getHeader()
 - getMimeType()
 - getName()
 - getSize()
 - getSurrogate()

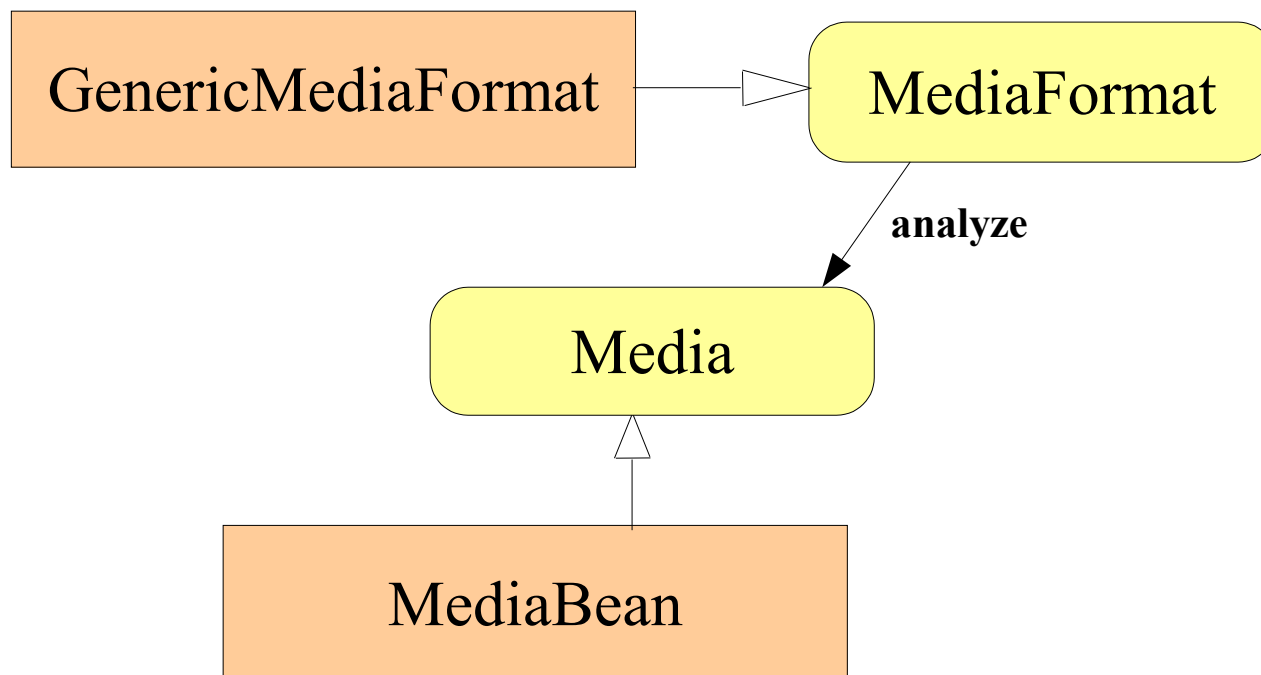
MediaBean

- Implémente Media
 - `MediaBean(InputStream contentStream, String mimeType, String name)`
 - `MediaBean(File mediaFile, String mimeType)`
 - `getContentStream()`

MediaFormat

- Interface qui étend Serializable
 - assembleContent(URL mediaLocation, MediaSegment[] mediaSegments)
 - disassembleContent(URL mediaLocation, byte[] mediaContent)
 - extractHeader(Media MediaObject)
 - extractSurrogate(Media MediaObject)
 - getDefaultMimeType()
 - isEmbedded()
 - isStreamingDesirable()

GenericMediaFormat



MediaFormatRegistry

- Instance unique
 - `MediaFormatRegistry()`
 - `bind(String fileExtension, MediaFormat)`
 - `getFileExtensions()`
 - `rebind(String fileExtension, MediaFormat mediaFormat)`
 - `unbind(String fileExtension)`
 - `lookup(String fileExtension)`

MediaSegment

- $\text{Media} := \{ \text{MediaSegment} \}$



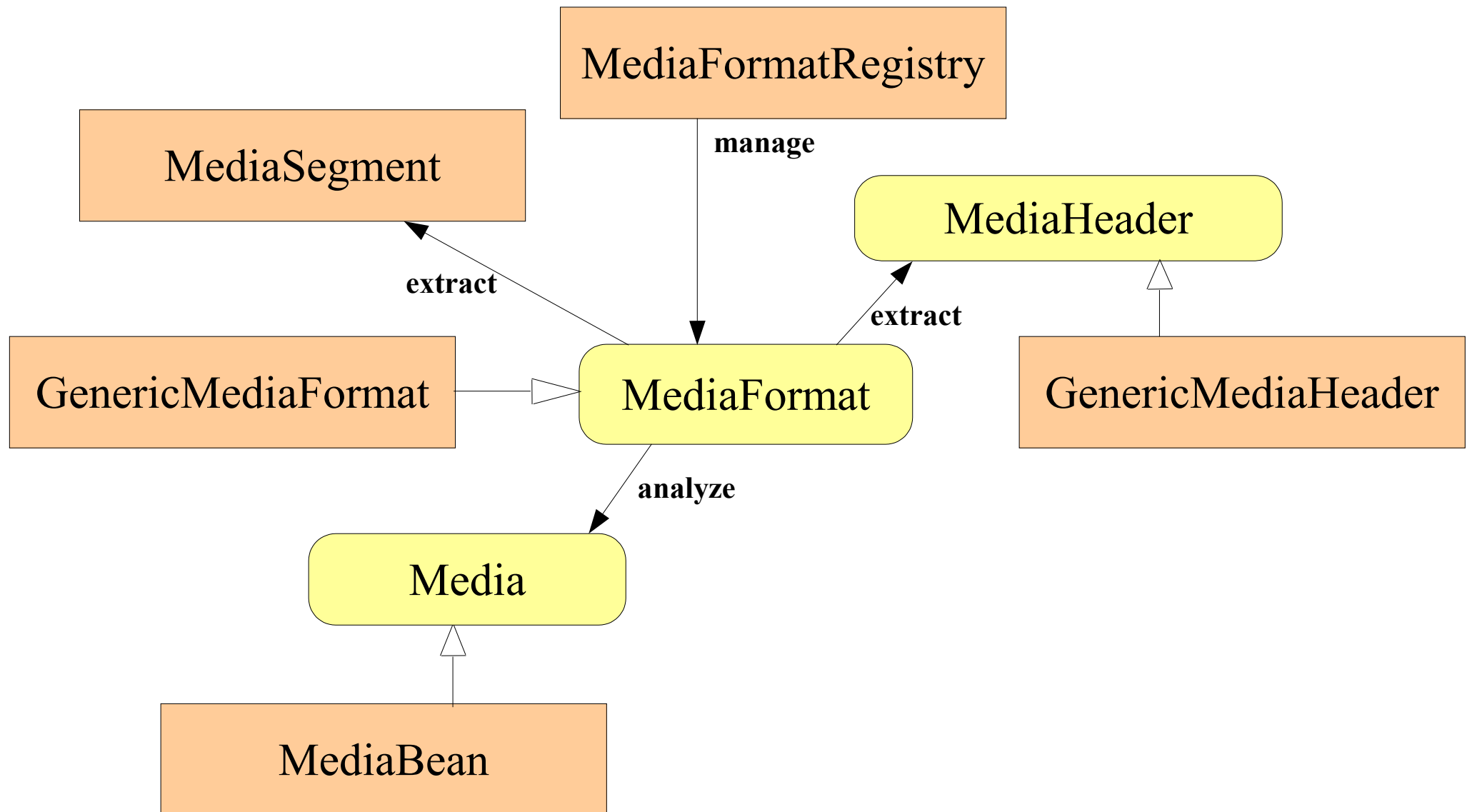
- $\text{MediaSegment} := \text{content} [\text{childLocation}]$
méthodes d'accès à **content**
méthodes d'accès à **childLocation**



MediaHeader

- Interface qui étend `Serializable`
 - méthodes d'accès aux champs du **Header**

GenericMediaHeader



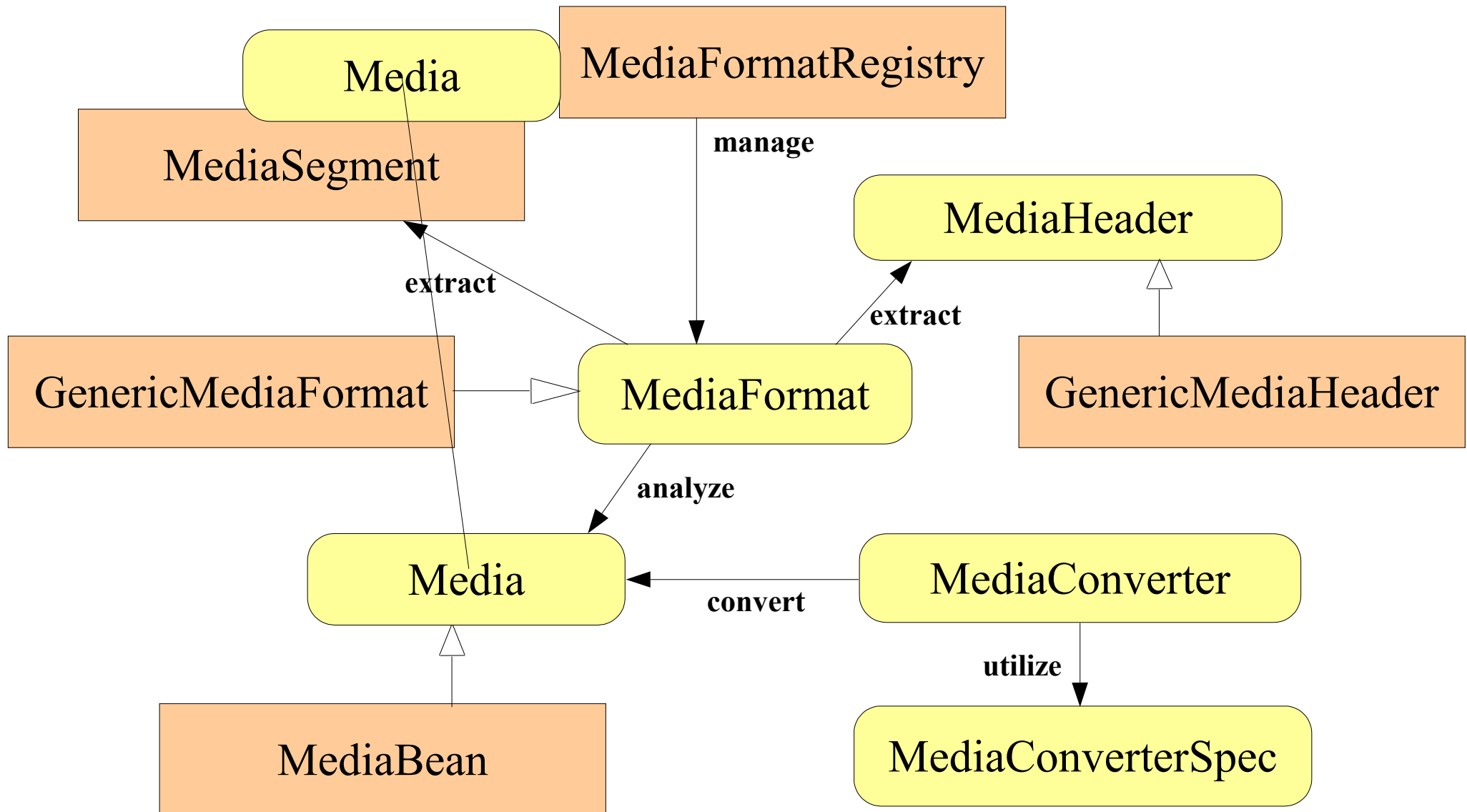
MediaConverter

- Interface
 - process(InputStream)
 - process(InputStream, OutputStream)

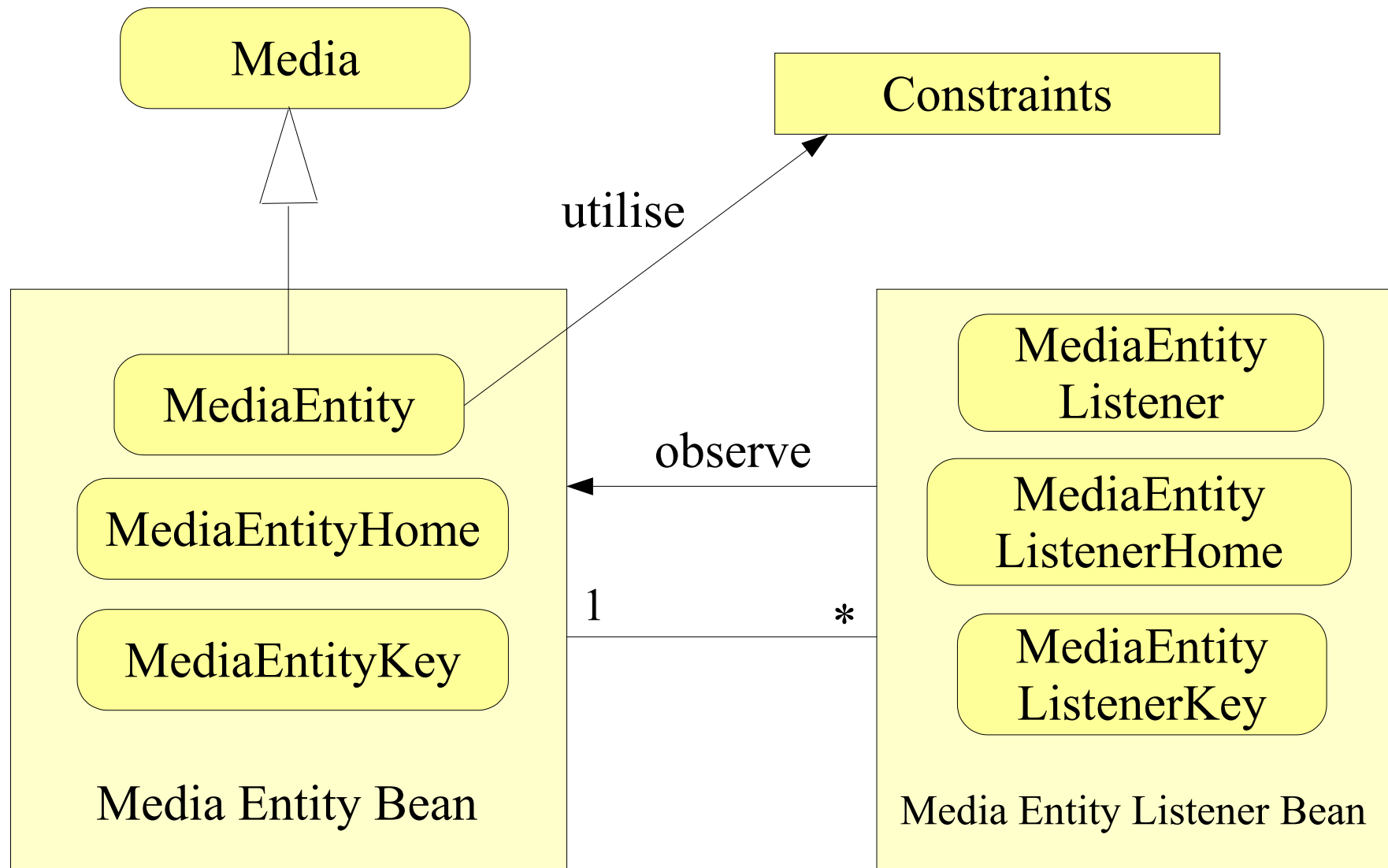
MediaConverterSpec

- Interface qui étend Serializable
 - `getConverter()`
 - `getTargetFileExtension()`
 - `getTargetMimeType()`

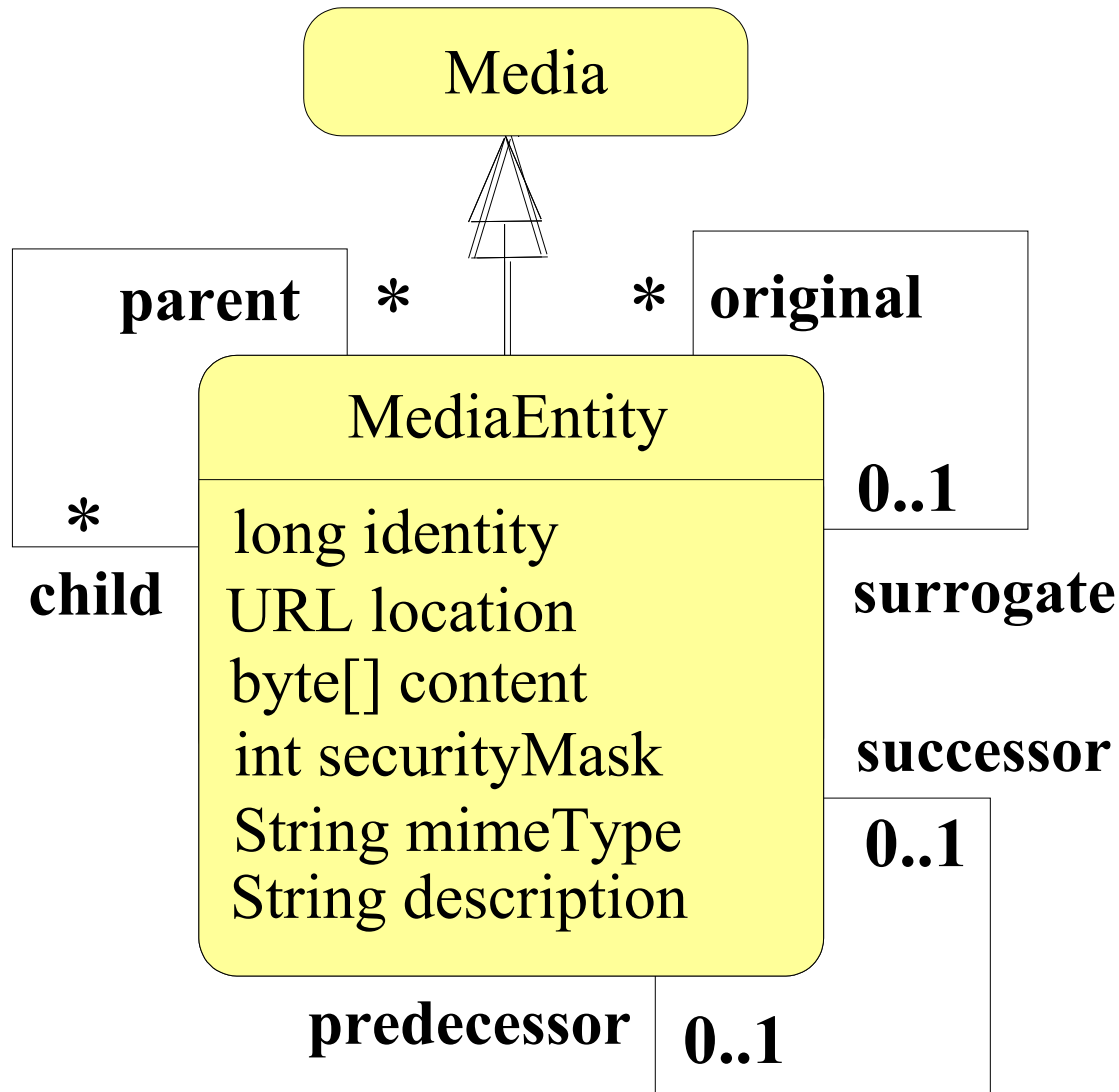
Media Entity Beans



Media Entity Beans



Media Entity



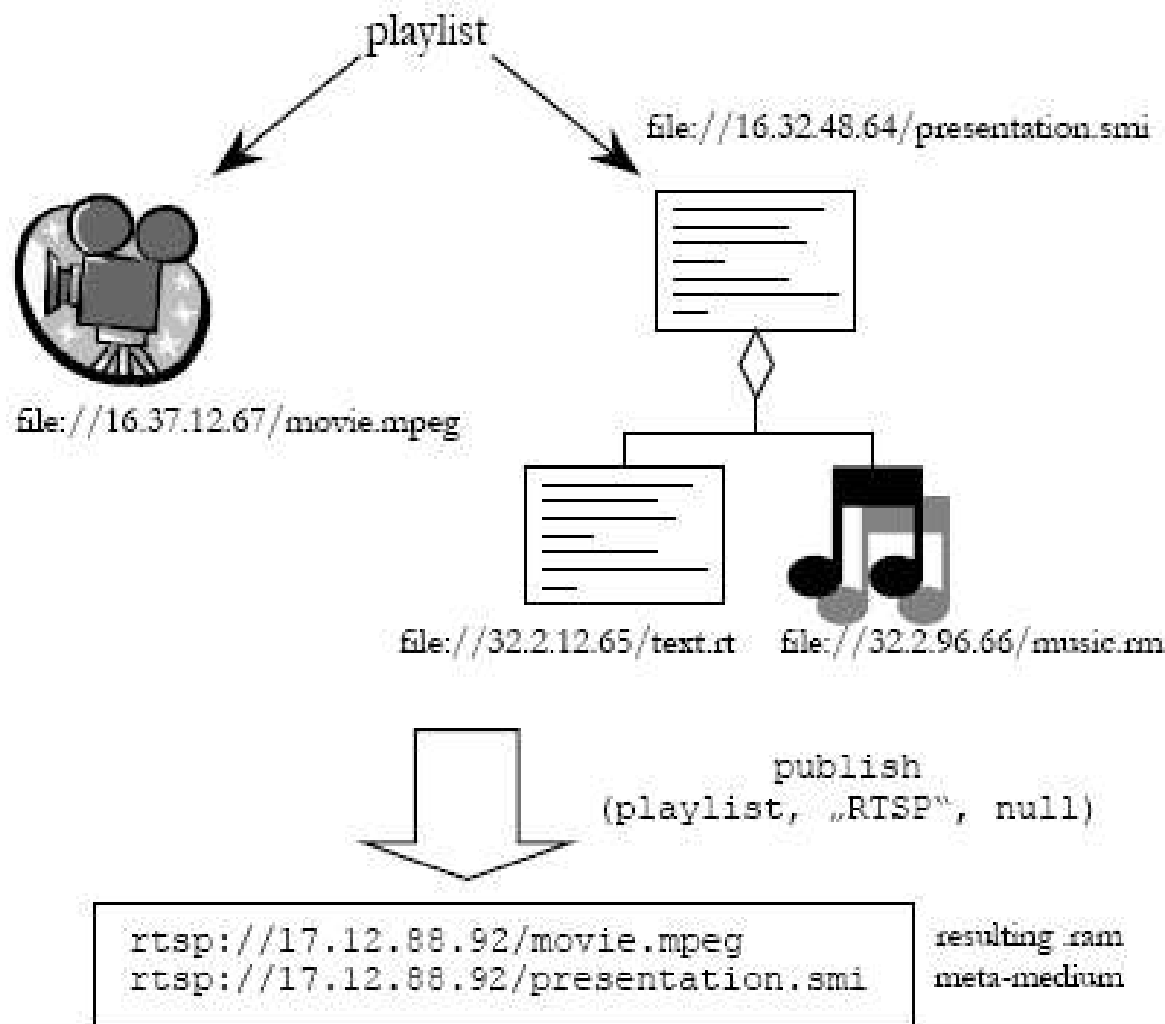
Media Entity

- `checkIn(URL sourceLocation)`
- `checkOut(URL targetLocation)`
- `convert(MediaConverterSpec[] specifications)`
- `getLastModified()`
- `stage()`

Media Entity Home

- des méthodes de recherche
- `polymorphicCreate(..., URL targetLocation)`
- `importMedia(URL[] sourceLocations, URL targetDirectoryLocation)`
- `exportMedia(...)`
- Deux méthodes `publish(...)`

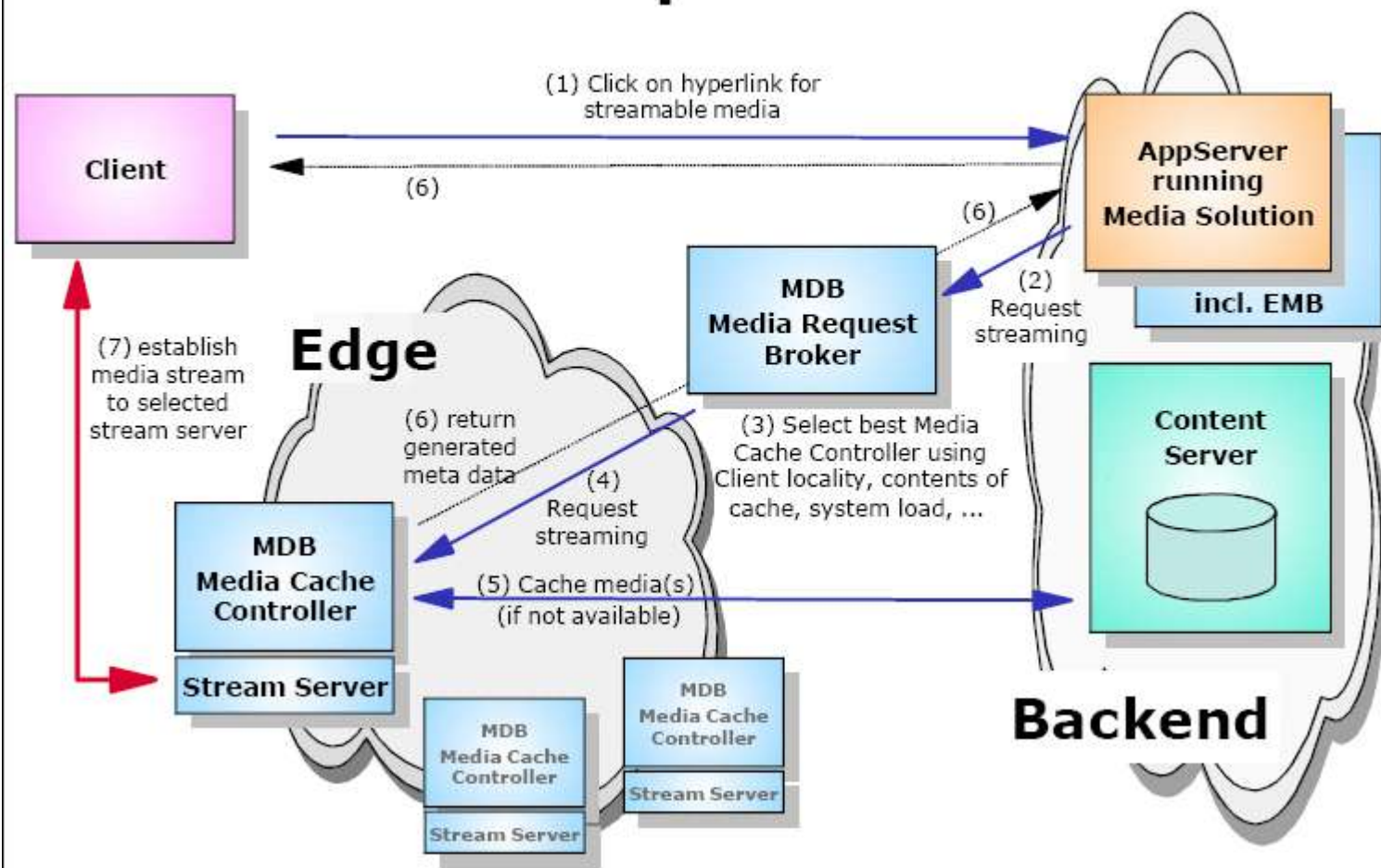
La méthode Publish



Sources : JSR086

Le Streaming

EMB Request flow



Sources : documentation du package RI d'IBM

Exemples de Programmation

- Obtenir une référence sur un proxy d'EMBHome

```
InitialContext context = new javax.naming.InitialContext();
```

```
mediaEntityHome = (MediaEntityHome) context.lookup("MediaEntity");
```

- Obtenir une référence sur un proxy d'EMB

```
MediaEntityKey mediaKey = new MediaEntityKey(Long.parseLong(identity);
```

```
MediaEntity mediaEntity = mediaEntityHome.findByPrimaryKey(mediaKey);
```

Exemples de Programmation

- Utilisation dans une servlet

```
MediaConverterSpec[] specs = new MediaConverterSpec[]
                                {new AddCopyrightConverterSpec();}
Media media = originalMedia.getConvertedMedia(specs);

long contentSize = media.getSize();

response.setContentLength((int) contentSize);
response.setContentType(media.getMimeType());

long position = 0;
while(position < contentSize) {
    byte[] partOfMedia = media.getContent(position,0x10000);
    response.getOutputStream().write(partOfMedia);
    position += partOfMedia.length
}
```


Exemples de Programmation

- Publication

```
URL link = mediaEntityHome.publish(mediaEntity,  
    MediaEntityHome.TRANSFERT_TYPE_BURST, null);
```

```
response.getOutputStream().write("<HTML><HEAD></HEAD><BODY><IMG SRC=\"");
```

```
response.getOutputStream().write(link.toExternalForm());
```

```
response.getOutputStream().write("\"></BODY></HTML> ");
```

Exemples de Programmation

- Publication

```
MediaEntity[] playlist = new MediaEntity[] {mediaEntity};
```

```
Media metaMedia = mediaEntityHome.publish(playlist,  
MediaEntityHome.TRANSFERT_TYPE_STREAM, null);
```

```
response.setContentLength((int) metaMedia.getSize());  
response.setContentType(metaMedia.getMimeType());
```

```
response.getOutputStream().write(metaMedia.getContent());
```

Conclusion

- Intégration des medias riches au sein de J2EE
- Facilite nettement leur manipulation
- Peu de serveurs J2EE
- Une petite partie implémentée

Perspectives

- MetaDataEntityLocal
- Utilisation des interfaces locales (EJB 2.0)
- Listener \neq Bean
- Intégré dans la plupart des serveurs J2EE

Documentation consultée

- **Enterprise Media Beans Specification, public draft,**
<http://www.jcp.org/aboutJava/communityprocess/review/jsr086/>
 - Référence pour les EMB
- **IBM, Programming Guide,**
<http://oss.software.ibm.com/developerworks/downloads/detail.php>
 - Permet de débiter
- **Alphaworks, FAQ,** <http://www.alphaworks.ibm.com/tech/emb>
 - Offre une vue globale et des news sur le sujet
- **IBM, EMB,**
<http://oss.software.ibm.com/developerworks/downloads/detail.php>
 - Package RI et TCK

Fin de la présentation

Merci pour votre attention

Des questions ?

Media Location

- Grammaire EBNF

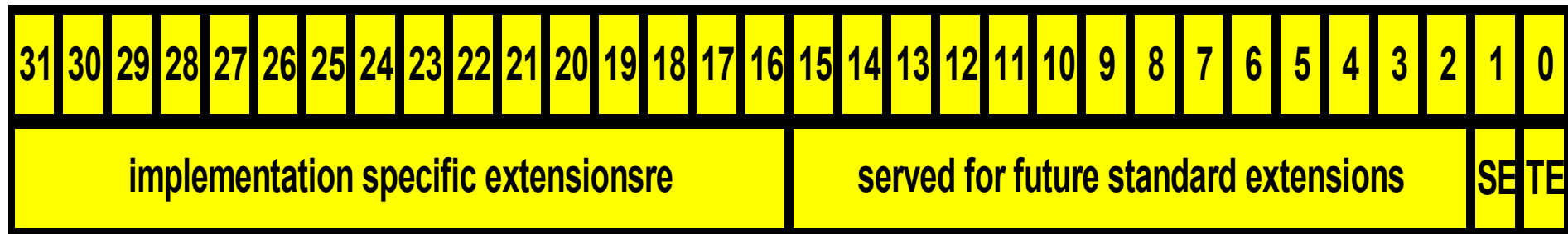
- Media location :=

- "file :/" <adresse IP> "/" <chemin et nom du fichier>

Formats supportés

Format	Extensions supportées	type MIME
MPEG Video	mpg, mpeg, mpv, mpegv, mpe, cda, vbs	video/mpeg
MOV	mov,qt	video/quicktime
AVI	avi	video/x-msvideo
ASF	asf	video/x-ms-asf
RealMedia	rm, rv, ra	application/vnd.rn-realmedia
MPEG Audio	mp3, mp2, mp1, mpa, mpga	audio/x-mpeg
WAV	wav	audio/x-pn-wav
JPEG	jpg, jpeg, jpe, pjp, pjpeg, jfif	image/jpeg
GIF	gif	image/gif
BMP	bmp	image/bmp
TIFF	tif, tiff	image/tiff
SMIL	smi, smil	application/smil
RealPix	rp	image/vnd.rn-realpix
RealText	rt	text/vnd.rn-realtex

Security Mask



- Transport Encryption
- Storage Encryption

Package RI

- Tests effectués par IBM
 - DB2 Universal Database 7.2 FP 9
 - WebSphere Application Server Base v 5.0.2
 - Windows 2000/XP
 - IBM HTTP Server (IHS) 1.3.26
 - WebSphere Studio Application Developer v 5.1

Démonstration

- Pas réalisable
- Utilisation de la RI d'IBM
 - Implique l'utilisation de DB2 et WAS