



Annotate Digital Video, Exchange on the Net

— Advene project

Le modèle Cinelab

Olivier Aubert
Pierre-Antoine Champin
Yannick Prié
LIRIS – Université Claude Bernard Lyon 1



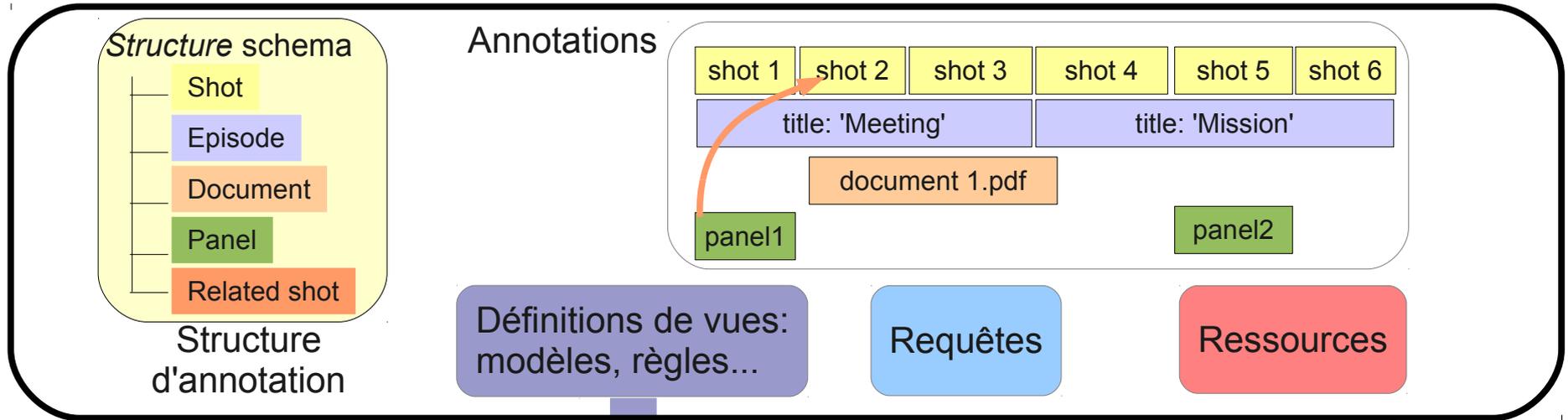
LIRIS – Laboratoire d'InfoRmatique en Image et Systèmes d'information
UMR 5205 – CNRS/INSA-Lyon/Université Lyon 1/Université Lyon 2/Centrale Lyon
<http://liris.cnrs.fr/>

- Advene : projet démarré en 2001
- Objectif : permettre la définition et l'échange de points de vue sur des vidéos
 - Hyper-documents liés à la vidéo
 - Lectures enrichies de la vidéo
- Importance d'un modèle générique
 - Émergence des usages
 - Partage et réutilisation

- Centré sur la notion d'**annotation** :
toute information (texte, son, image...)
attachée à un fragment de la vidéo
- Séparation des niveaux :
 - Annotation
 - Structuration (Relations, Types, Schémas)
 - Présentation (Vues, Requêtes)

- Les méta-données peuvent être partagées, échangées, réutilisées
 - Regroupées dans un **recueil** incluant les trois niveaux (annotation, structure, présentation)
 - Possibilité d'import dynamique (référence) entre les recueils
 - Le recueil *référence* uniquement la vidéo (partage, copyright)

Recueil Advene




Limitations initiales

L'implémentation et le modèle initial d'Advène présentent quelques limitations :

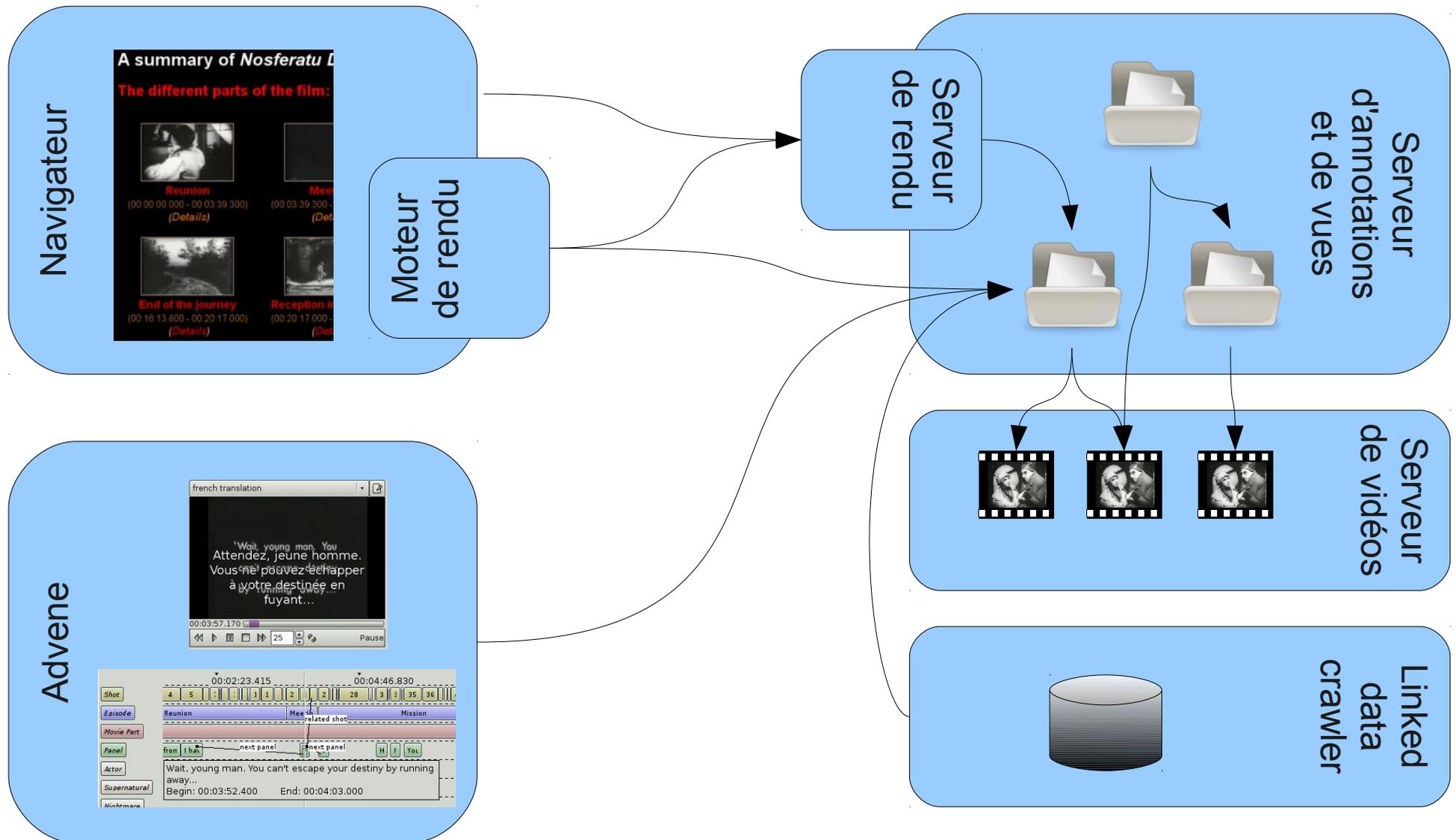
- Chaque recueil est en pratique limité à une seule vidéo
- Structuration insuffisante
- Imports difficiles à gérer
 - Limite le partage et la réutilisation

Projet ANR Cinelab (2007-2008)

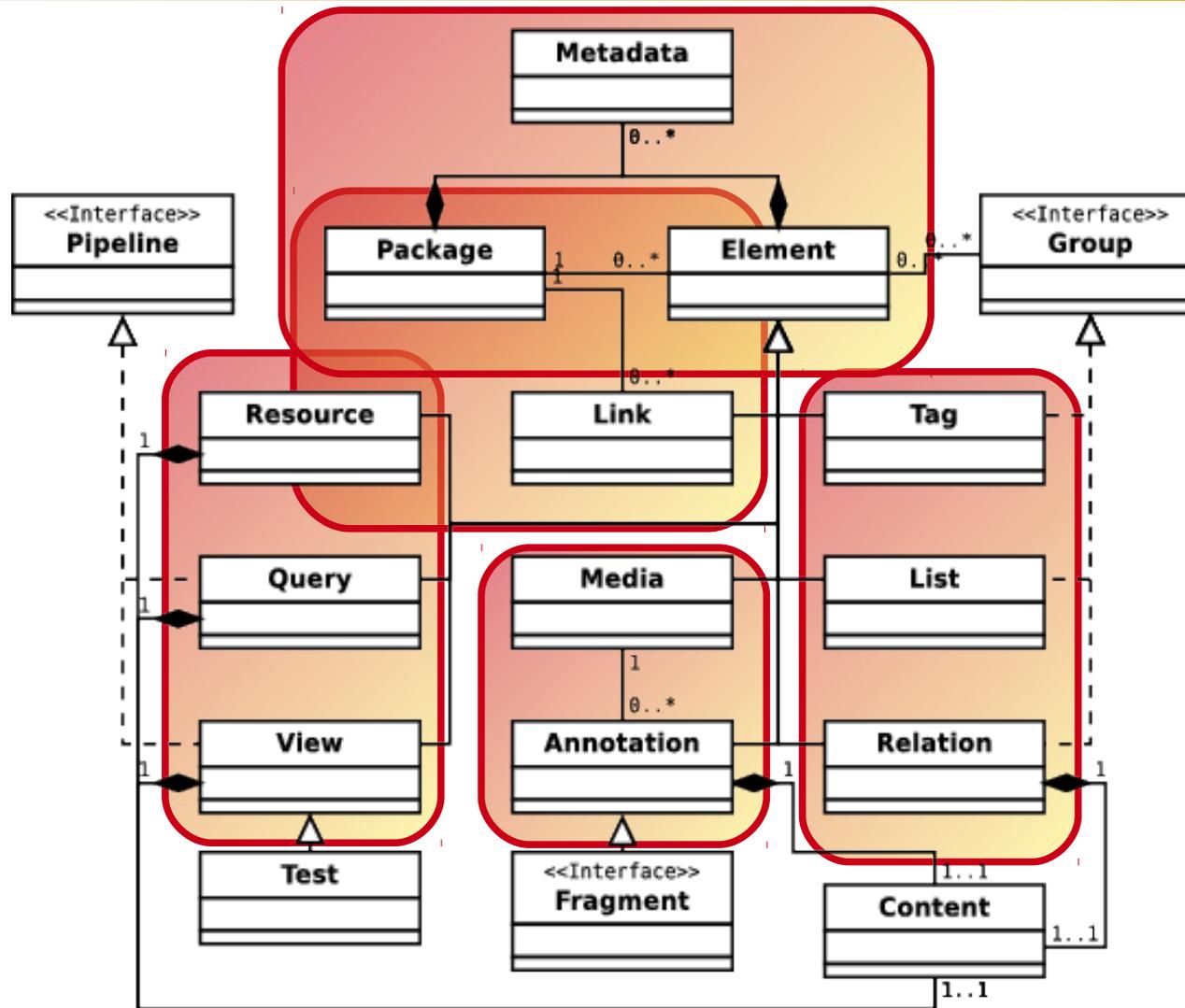
- Évolution du modèle pour
 - Pallier les limitations rencontrées
 - Répondre aux besoins des partenaires
- Nouvelle implémentation
 - Éliminant certaines erreurs de jeunesse
 - Plus modulaire

- Représentation explicite des vidéos
- Structuration plus souple
 - Liste (ordonnée, fermée)
 - Tag (sans ordre, ouvert)
- Imports de recueils entiers
- Méta-données mieux formalisées, permettant l'extension du modèle
 - Émergence

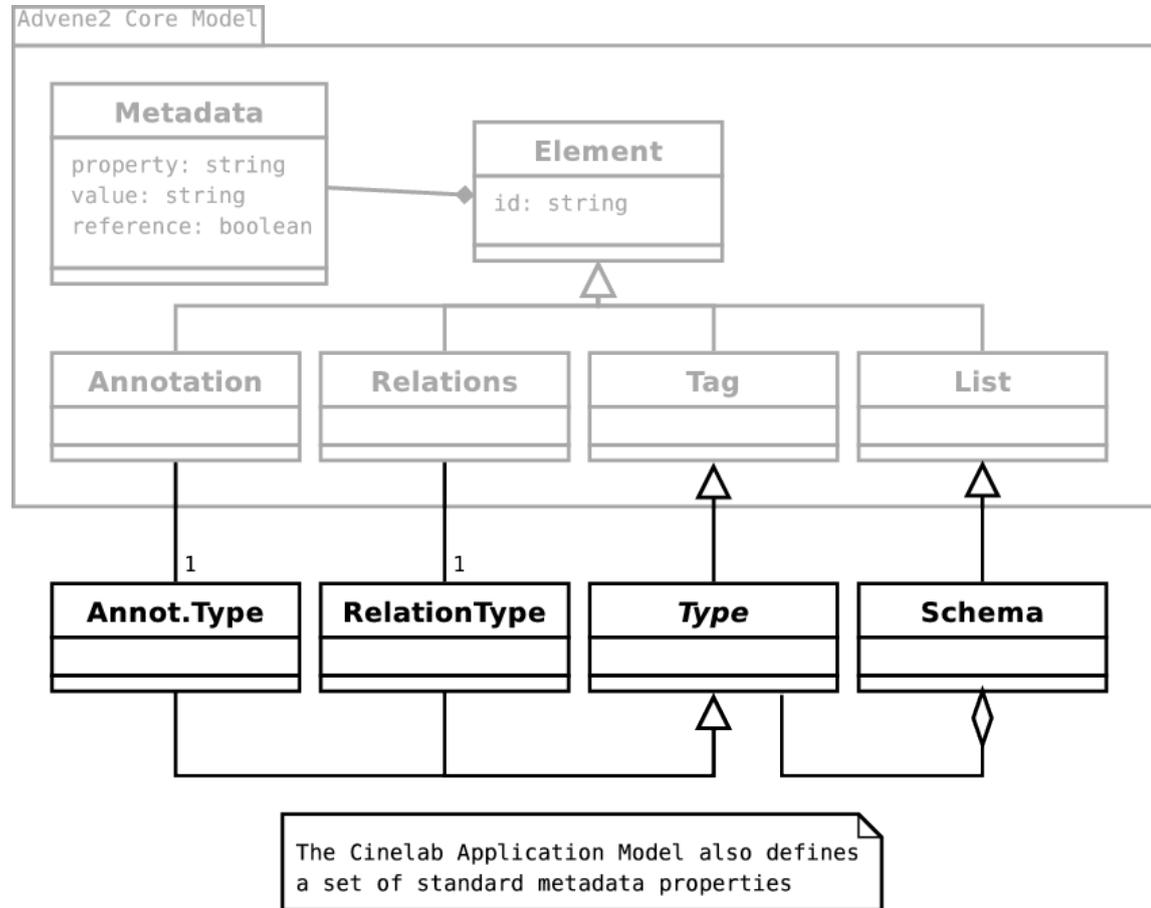
Architecture globale



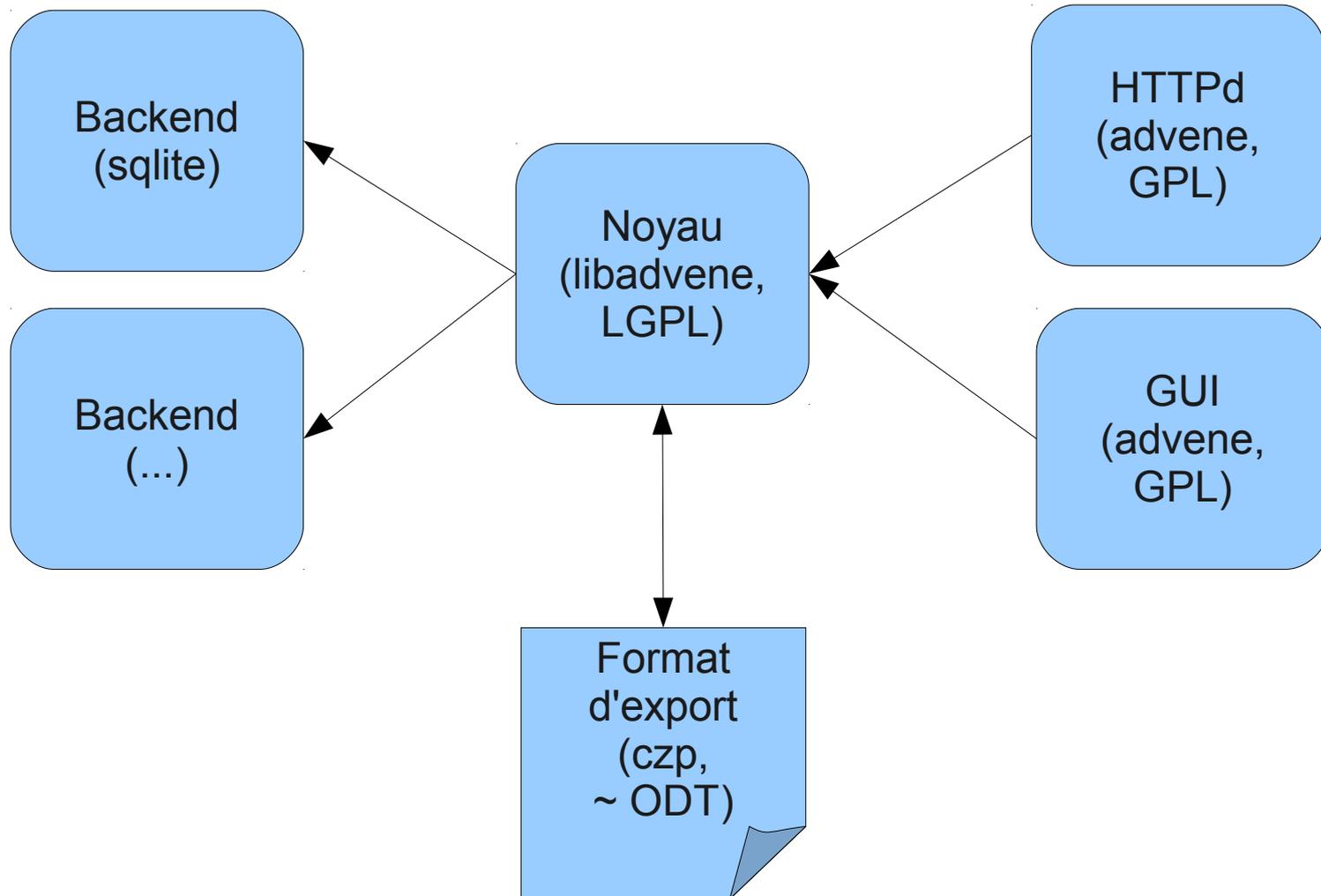
Aperçu du modèle noyau



Aperçu de la couche applicative



Aperçu de l'implémentation



```
<package xmlns="http://advene.org/ns/cinelab/">
  <meta>...

  <annotation-types>
    <annotation-type id="my_notes">
      <meta>...

  <medias>
    <media id="my_video"
      url="/home/pa/perso/video/video.mp4"
      origin="0" unit="ms"
      <meta>...

  <annotations>
    <annotation id="a1"
      media="my_video" begin="0" end="1000" >
      <content mimetype="text/plain">
        Hello world
      <meta>
        <type id-ref="my_notes" /> ...
```

Contenu des balises méta (requis pour tous les éléments du modèle applicatif Cinélab) :

```
<meta xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:creator>
    pa
  <dc:created>
    2009-09-16T17:06:11.830640
  <dc:contributor>
    pa
  <dc:modified>
    2009-09-16T17:10:52.392980
  ...
</meta>
```

- MPEG7 : par import/export
- OAI-PMH : implémentation possible dans le serveur web

- Le modèle est solide (évolution de Advene1 et prise en compte des problèmes rencontrés depuis 8 ans)
- Une implémentation existe
 - Python
 - Portable (Windows, MacOS, Linux)
 - Open Source (LGPL)

Annexe 1 : modèles de contenu (1)

- Tout contenu possède un type *mime* qui permet :
 - de fournir une interface adaptée à sa saisie
 - de vérifier sa validité syntaxique
 - de l'interpréter
- Mais certains types mimes peuvent être contraints plus précisément (e.g. XML).

Annexe 1 : modèles de contenu (2)

- Tout contenu peut être associé à une ressource définie comme son *modèle de contenu*.
- Le modèle cinélab préconise les modèles de contenus suivants :
 - Perl Compatible Regular Expression
 - Relax NG (compact syntax)
 - JSON-schema

Annexe 1 : couples attribut-valeur

- Besoin exprimé : liste de couples attribut/valeur dont
 - les attributs sont pré-définis
 - certains attributs prennent leur valeur dans un vocabulaire contrôlé (thésaurus)
- Implémentation proposée :
 - structure : sous-ensemble de Relax NG
 - thesauri : SKOS
 - lien entre les deux : annotations RNG

Annexe 1 : exemple (Relax NG)

```
element shot_number { xsd:int },  
element title { text },  
element description {text}?,  
  [ thesaurus="th_angle" ]  
element angle { "front" | "high" | "low" },  
  [ thesaurus="th_characters" ]  
element characters {  
  "jonathan" | "nina" | "nosferatu" |  
  "reinfield" | ...  
}*
```

Annexe 1 : exemple (SKOS)

@prefix : <http://www.w3.org/2004/02/skos/core#> .

<PrimaryCharacter> a :Concept ;
:prefLabel "Primary Character"@en, "Personnage principal"@fr.

<jonathan> a :Concept ;
:broader <PrimaryCharacter> ;
:prefLabel "Jonathan Harker".

<nina> a :Concept ;
:broader <PrimaryCharacter> ;
:prefLabel "Nina".

...

<SecondaryCharacter> a :Concept ;
:prefLabel "Secondary Character"@en, "Personnage secondaire"@fr .

<inn-keeper> a :Concept ;
:broader <SecondaryCharacter> ;
:prefLabel "Inn keeper"@en, "Aubergiste"@fr .

....

Annexe 1 : remarques

- Redondance entre le RNG et le SKOS
 - le RNG est auto-suffisant
 - on peut automatiser la synchronisation
- Avantages de l'ajout de SKOS :
 - meilleures interfaces, exploitant la hiérarchie, les libellés
 - meilleure exploitation (e.g. « Trouve tous les plans contenant un personnage principal »)