Programmation XML

ENPC - Module SYSIN - Option XML

Bernd Amann

Objectifs du cours

- Comprendre l'utilisation XML dans le contexte d'une application Web.
- Apprendre la syntaxe XML et le(s) modèle(s) sous-jacent(s).
- Etudier et mettre en oeuvre quelques outils (langages) pour la manipulation (programmation) de XML.

Plan du cours

- Introduction: Le rôle de XML dans les applications Web
- XML: syntaxe, modèles DOM et SAX, DTD
- XPath : extraction de fragments XML
- XSLT: transformation de documents XML

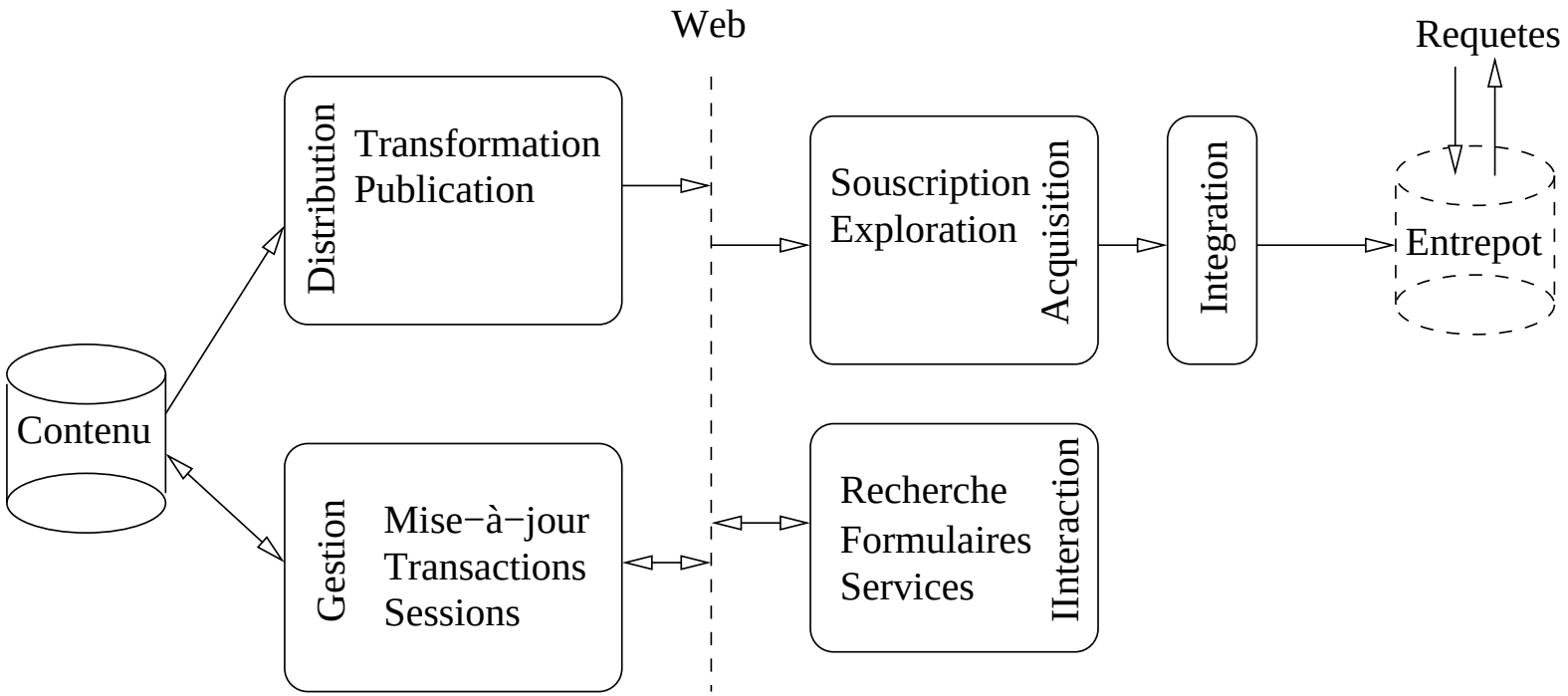
Introduction: Le rôle de XML dans les applications Web

Applications Web

- Commerce électronique : catalogue/achat en-ligne (B2C), échanges commerciales (B2B)
- Éducation : formation/enseignement à distance
- Recherche : organisation de conférences, journaux électroniques
- Médias : information en directe, archivage
- Archivage, veille technologique (entrepôts thématiques)

Problème commun : Comment échanger et gérer des informations sur le Web ?

Architecture d'une application Web



Défis d'une application Web

Le Web est un environnement **distribué** de ressources, serveurs, applications, clients, etc.. **autonomes** à l' **grande échelle** (taille et nombre de ressources, nombre de clients, ...)

Problème: Prendre en compte toutes les dimensions du Web qui sont souvent contradictoires.

Exemple: Moteur de Recherche

Un problème majeur rencontré par les moteurs de recherche est de traiter des milliards de pages avec des ressources (mémoire, bande passante, ...) limitées.

Maintenance de l'index :

- acquisition de nouvelles pages
- rafraîchissement des pages

Solutions :

- modèle de coût basé sur l'importance d'un page, son âge, ...
- publication/souscription

Exemple: Moteur de Recherche

L'évaluation de requêtes :

- nombre d'utilisateurs, taille des données
- taille du resultat
- temps de réponse

Solutions :

- index en mémoire
- évaluation incrémentale (flux)
- mésures d'importance des pages

Exemple: Entrepôt de données

L'autonomie des ressources (et des personnes qui les produisent) a comme résultat une hétérogénéité à différents niveaux.

- Format et structure : l'information est représentée sous différents formats et structures
- Connaissances : l'information est produite et interprêtée avec des connaissances/contextes différents.
- Outils : l'information est gérée et produite par des outils différents.

Problème : On est obligé d'adapter *les données aux applications.*

Composants d'une application Web

Pendant le développement d'une application Web il faut prendre en compte :

- les protocoles : HTTP, SMTP, RMI, Corba, telnet, ...
- les modèles de données : HTML, Java, relations, ...
- les environnements de programmation : Servlets, scriptes CGI, PHP, ...

Problème : Je voudrais intégrer n applications sans gérer n^2 interfaces.

Évolution du traitement de l'information

Système de Fichiers :

- stockage et traitement sont fortement liés
- modèle physique = modèle logique

SGBD :

- architectures client-serveur
- séparation entre modèle physique et modèle logique (relations)

Web :

- architectures à trois niveaux (three-tier): entrepôts et médiateurs
- architectures pair-à-pair (P2P): services Web
- séparation entre modèles "source" et modèle "global": XML

XML comme modèle de données

XML est LE modèle de données pour le Web.

Un document XML

- peut être échangé facilement (format ASCII) ;
- permet de représenter pratiquement toute information structurée ;
- n'est pas lié à un mode d'utilisation : chacun peut se définir ses propres « structures » ;
- peut être stocké, transformé, interrogé "facilement"

Gestion de Contenus Web : le rôle de XML

La technologie XML (dans le sens très large) intervient à tous les niveaux dans les applications (Web) :

- Stockage de données : entrepôts et BD XML
- Manipulation de données : XPath/XQuery (interrogation), DOM/SAX (programmation), XSLT (transformation), ...
- Communication entre applications : services Web (SOAP)
- Publication de données : XHTML, SMIL, SVG, ...
- Web Sémantique : RDF

Une étude de cas

L'Officiel des spectacles !

- Une base de données avec des films
- Des salles de cinéma, avec des séances de projection de films
- Des cinémas, qui diffusent leur programme sur le Web, sur le WAP, sur des tracts et des affiches...
- Un moteur de recherche pour chercher des séances, des films, des horaires

La fiche du film *Gladiator*

La fiche du film peut être publiée

- en HTML pour Netscape/IE
- en WML pour le portables WAP
- en SMIL pour Realplayer
- dans un moteur de recherche SallesEnLigne.com

L'information ?

- c'est la *même*, sous des formes différentes
- elle est échangée entre plusieurs acteurs

Les solutions XML ?

- **Format universel** :
  - représentation en chaîne de caractères d'un contenu structuré
  - indépendant de toute application
- **Publier l'information**
  - outils de transformation simples pour convertir un contenu XML
- **Échanger et intégrer l'information**
  - assembler des contenus XML, ou au contraire en extraire des informations

XML, format universel

XML, c'est quoi ?

XML = rendre un **contenu** accessible à toute application.

Le contenu :

```
L'Epée de bois, 100 rue Mouffetard, métro
Censier-Daubenton
```

Le même, en XML :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CINEMA><NOM>Epée de Bois</NOM><ADRESSE>100,
rue Mouffetard</ADRESSE><METRO>
Censier-Daubenton</METRO></CINEMA>
```

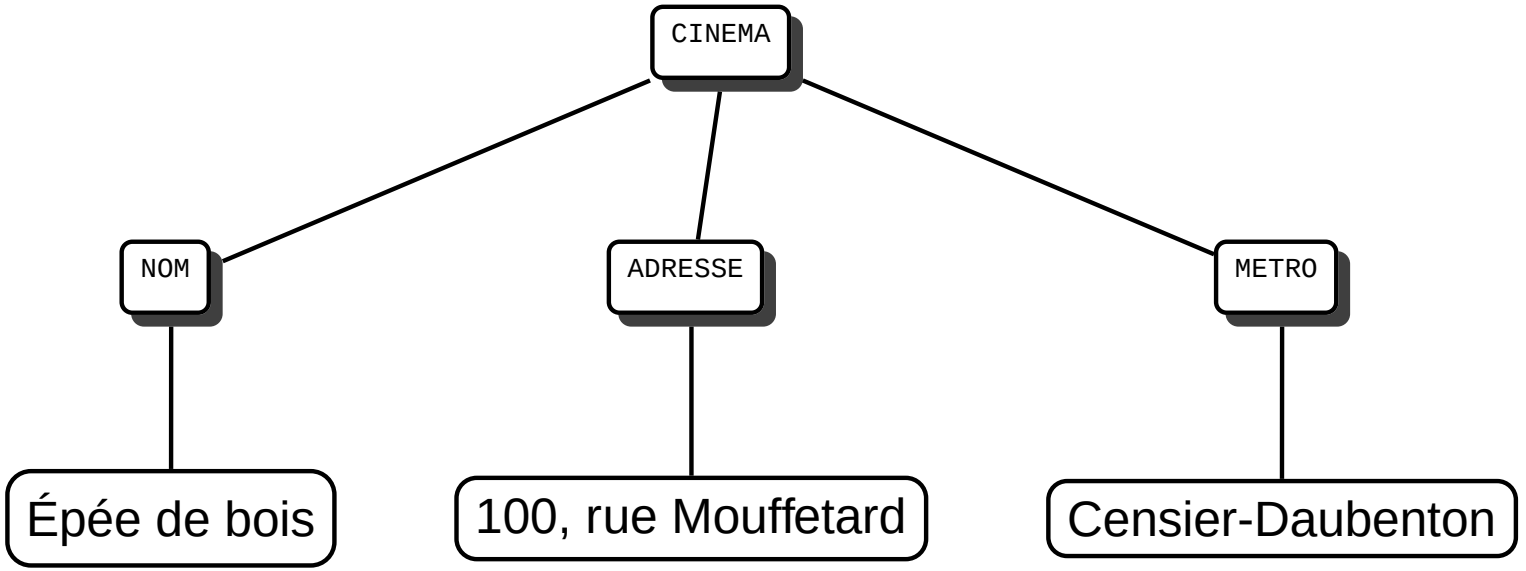
Le même, mieux présenté

Présentation courante : avec indentation

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CINEMA>
  <NOM>
    Epée de Bois
  </NOM>
  <ADRESSE>
     100, rue Mouffetard
   </ADRESSE>
  <METRO>
     Censier-Daubenton
  </METRO>
</CINEMA>
```

NB : il y a des espaces et des sauts de ligne

Encore mieux : sous forme d'arbre



Documents XML

Qu'est-ce qu'un **document XML** ?

- C'est un **contenu** alphanumérique
- Il est **structuré** avec des balises

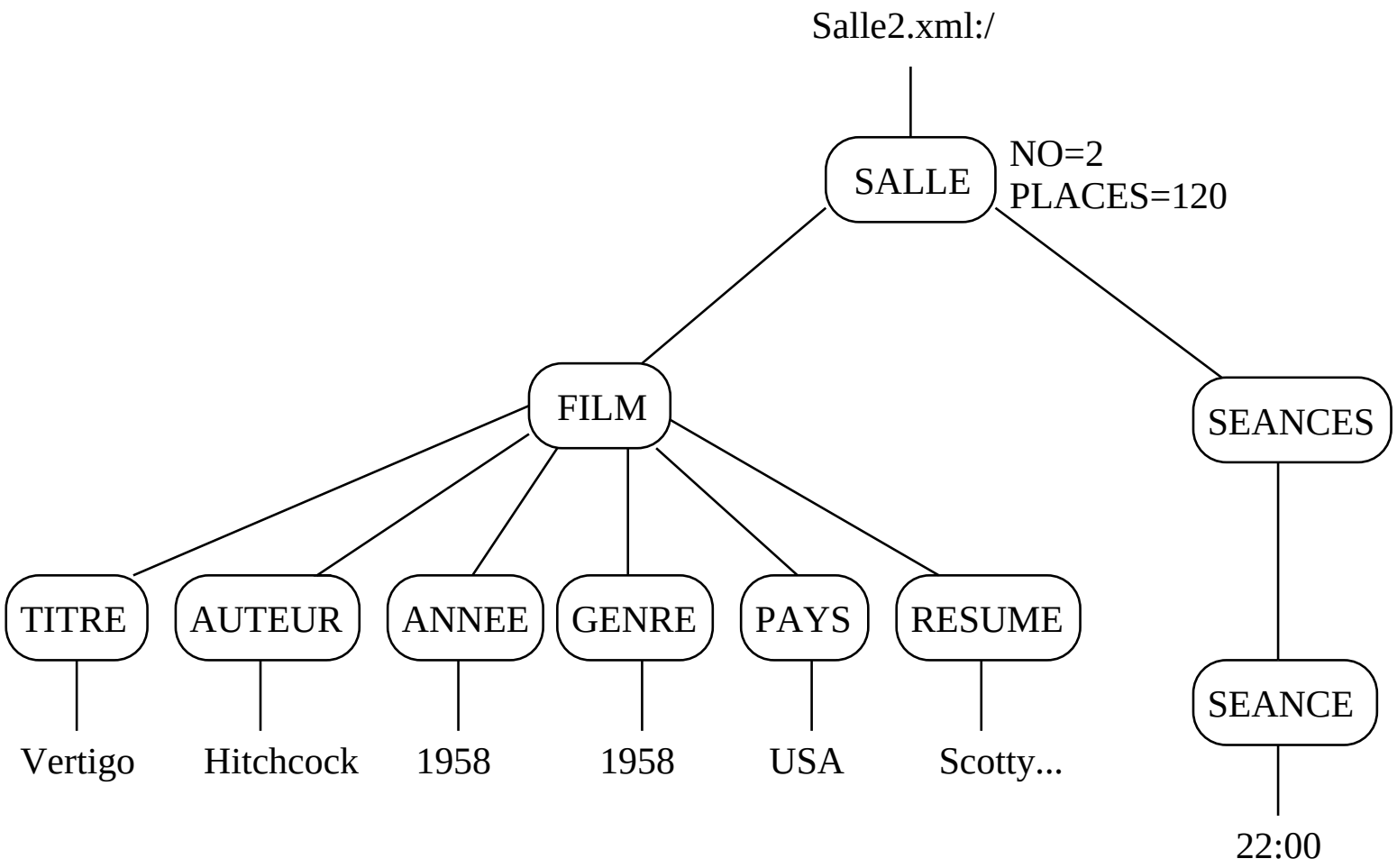
Indépendant de la **représentation physique** (origine)

- Un ou plusieurs fichiers ?
- Un message ?
- Un extrait d'une base de données ?
- Tout ça à la fois ...

Intégration : les salles

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SALLE NO='2' PLACES='120'>
 <FILM>
  <TITRE>Vertigo</TITRE>
  <AUTEUR>Alfred Hitchcock</AUTEUR>
  <ANNEE>1958</ANNEE>
  <GENRE>Drame</GENRE>
  <PAYS>Etats Unis</PAYS>
  <RESUME>Scottie Ferguson, ancien inspecteur
       de police, est sujet au vertige depuis
       qu'il a vu mourir son collègue....
  </RESUME>
 </FILM>
 <SEANCES>
    <SEANCE>22:00</SEANCE>
 </SEANCES>
</SALLE>
```

Sous forme d'arbre



Le cinéma : intégration des salles

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE CINEMA [
  <!ENTITY salle1 SYSTEM "Salle1.xml">
  <!ENTITY salle2 SYSTEM "Salle2.xml">
]>
<CINEMA>
  <NOM>Epée de bois</NOM>
  <ADRESSE>100, rue Mouffetard</ADRESSE>
  <METRO>Censier-Daubenton</METRO>
  &salle1;
  &salle2;
</CINEMA>
```

Exemple: Publication de données avec XML

Publication de documents XML

Objectif: **Séparer** la gestion du contenu de la présentation

- Gestion du contenu => décrire nos informations, avec un vocabulaire XML
- Présentation => mettre en forme nos documents pour une application particulière

Le langage XSLT permet d'écrire des programmes de conversions, très adaptés au traitement de documents XML

Application Cinéma

Nous avons décrit notre cinéma avec notre propre format XML.
XSLT va permettre de traduire ce langage vers d'autres formats :

- HTML pour la présentation Web standard
- WML pour la présentation WAP
- SMIL pour une présentation multimédia
- XSL-FO pour la production de documents papier

Version HTML

HTML revisité :

- Un document HTML **est** un document XML
- Le vocabulaire est fixé, ainsi que la syntaxe
- Chaque balise a une signification bien définie

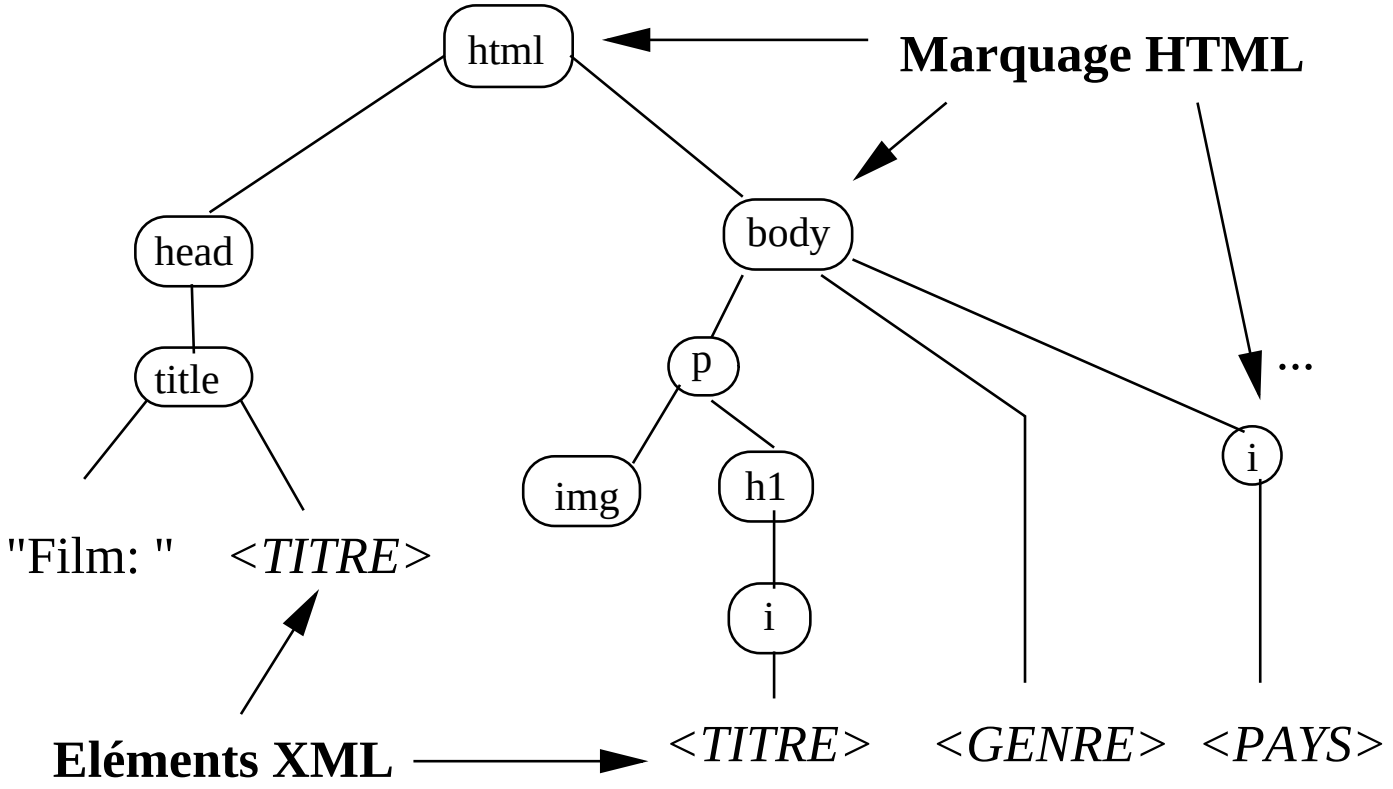
HTML a été normalisé comme « dialecte » XML
=> c'est XHTML

Ce qu'on veut obtenir

(démo)

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
     <title>Film: Vertigo</title>
   </head>
   <body bgcolor="white">
      <p><img SRC="Vertigo.gif" align="left" height="220">
         <h1><i>Vertigo</i></h1>Drame, <i>Etats Unis</i>, 1958
      </p><p>
         Mis en sc&egrave;ne par  <b>Alfred Hitchcock</b>
        <h3>R&eacute;sum&eacute;</h3>Scottie Ferguson,
            ancien inspecteur de police, est sujet
            au vertige depuis qu'il a vu mourir son
            coll&egrave;gue. Elster, son
            ami, le charge de surveiller sa femme, Madeleine, ayant des
            tendances suicidaires. Amoureux de la jeune femme Scottie ne
            remarque pas le pi&egrave;ge qui se trame autour de lui
            et dont il va &ecirc;tre la victime...
      </p>
   </body>
</html>
```

HTML, sous forme d'arbre



Le rôle de XSLT

XSLT permet :

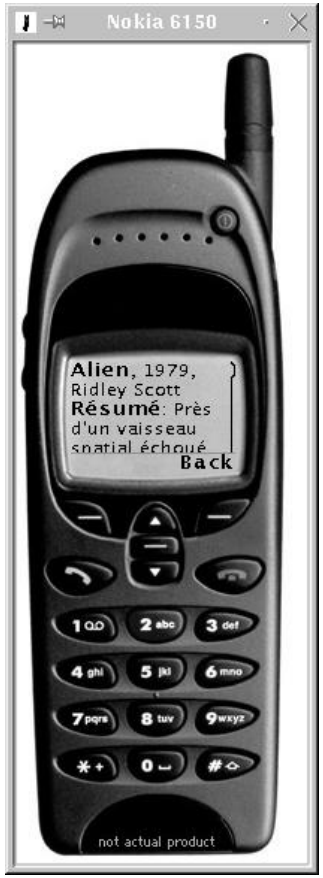
- De prendre en entrée un **document XML source**
- De produire en sortie un autre arbre XML
- D'insérer dans le document en sortie des fragments du document source

Donc bien adapté à une transformation XML -> HTML

WML, autre dialecte de XML

- Document WML : marqué par la balise `<wml>`
- Il est divisé en *cartes*, unité d'affichage sur le mobile (`<card>`)
- Elements principaux :
  - des balises simples de mise en forme (`<b>`, `<i>`)
  - des *ancres* pour passer d'une carte à une autre

Exemple d'une carte WML




```
<?xml version="1.0"
      encoding="iso-8859-1"?>
<wml>
  <card>
    <p>
      <b>
        Alien
      </b>
      , 1979, Ridley Scott
    <br/>
    Près d&apos;un vaisseau spatial
    échoué sur une lointaine
    planète, ..
   </p>
  </card>
</wml>
```

Un site WAP

On envoie un **ensemble** de cartes :

- Dotées d'une **identité** :

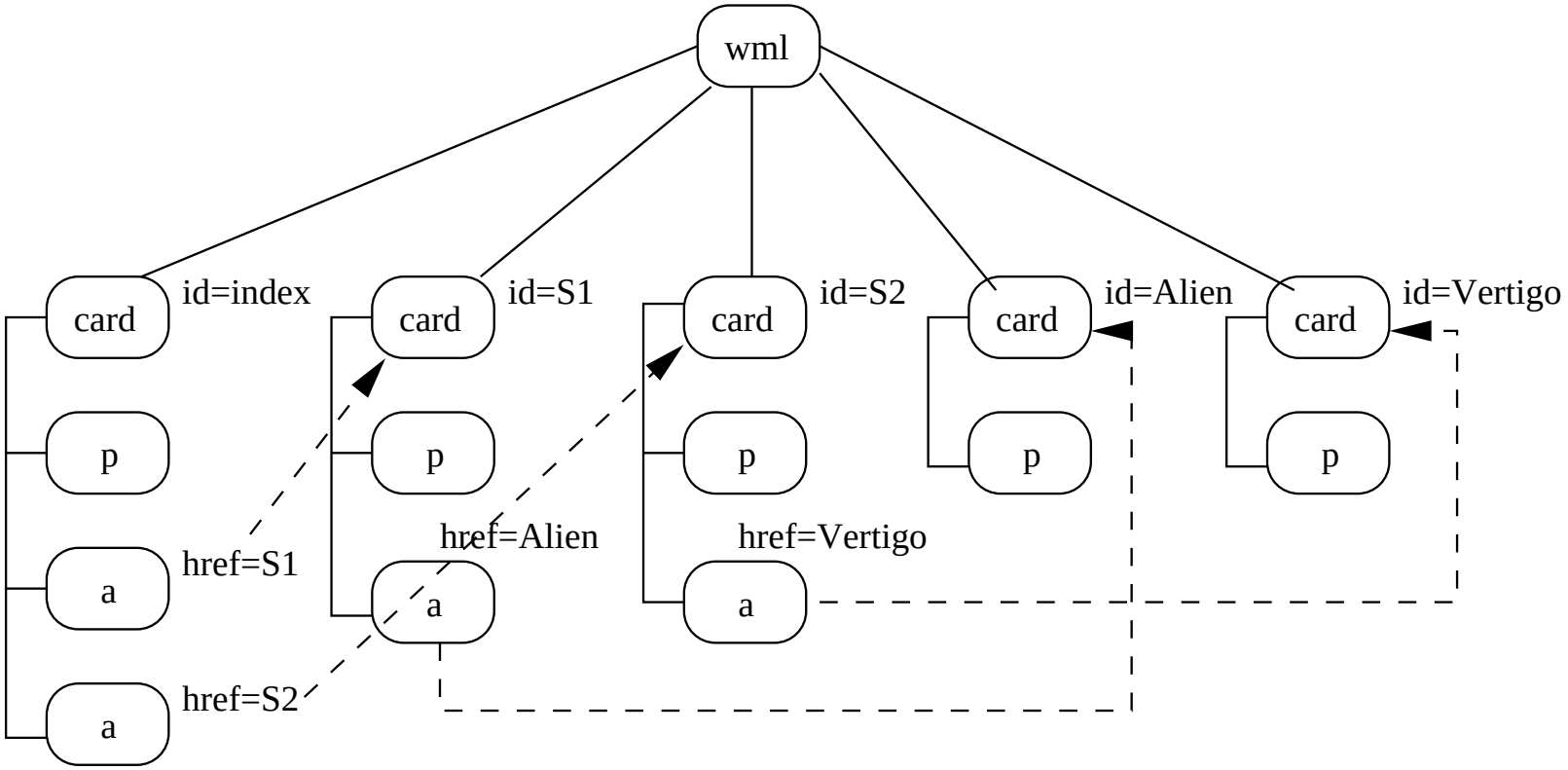
```
<card id="Alien"> ...  suite de la carte
</card>
```

- **Référençant** d'autres cartes :

```
<a href="#Alien">lien vers la carte Alien</a>
```

Les cartes sont « compilées » et transmises par le réseau sans fil.

Arbre XML du site WML



Le document

```
<wml>
  <card id="index" title="Programme">
      ...
      <a href="#S1"> Salle 1: </a>
      ...
  </card>
  <card id="S1">
      S&#xE9;ances salle 1  <p>
      <a href="#Alien"> Film : Alien</a>
      ...
  </card>
  <card id="Alien">
      ...
  </card>
</wml>
```

SMIL

Langage pour la création de documents **multimédia** avec XML.

- On indique une fenêtre d'affichage avec différentes régions pour l'affichage des composants.
- On place les composants dans les différentes régions (positionnement spatiale).
- On synchronise l'affichage des composants (positionnement temporel).

Structure d'un document SMIL

```
<smil>
  <head>
    <meta ... />          <!-- infos   -->
    <root-layout .../> <!-- fenêtre -->
    <region ... />        <!-- région 1-->
    <region ... />        <!-- région 2-->
  </head>
  <body>
                          <!-- contenu -->
  </body>
</smil>
```

Le corps contient deux catégories d'éléments:

- objets multimedia: `<text>`, `<image>`, `<audio>`, `<video>`, `<textstream>`
- éléments de synchronisation: `<seq>` (séquence), `<par>` (groupe "parallèle")

Exemples de composants textuels

- Vertigo-Title.rt:

```
<window type="marquee" height="50" width="200" bgcolor="black">
   <font size="50" face="Garamond" color="red">Vertigo toto</fon
</window>
```

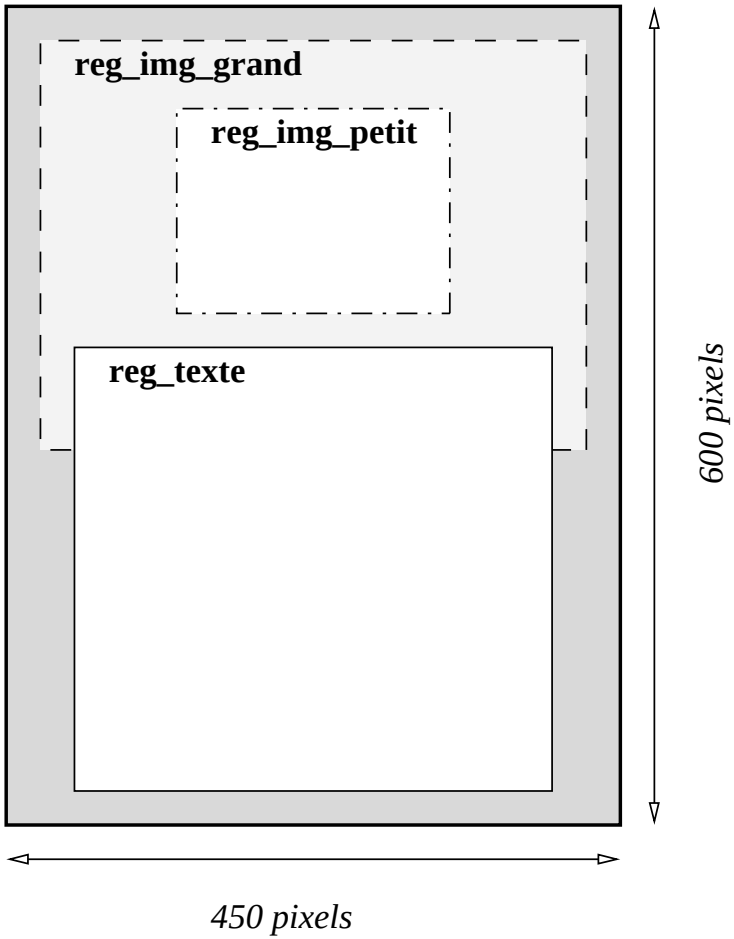
- Vertigo-Info.rt:

```
<window type="marquee" height="50" width="200" bgcolor="black" l
   <font color="white">de Alfred Hitchcock, Drame, Etats Unis, 1
</window>
```

Exemple SMIL: Fenêtre et régions

```
<head>
 <layout>
  <root-layout width="300" height="400"/>
  <region id="img_grand" width="300"
          height="400" fit="meet"/>
  <region id="img_petit" width="150"
          left="80" height="100" top="10"/>
  <region id="txt1"    width="200"
          left="50" height="50"  top="120"/>
  <region id="txt2"    width="200"
          left="50" height="50"  top="180"/>
  <region id="txt3"    width="250" left="25"
          height="150" top="240"/>
 </layout>
</head>
```

Exemple SMIL: Fenêtre et régions



Exemple SMIL: Positionnement

```
<body><seq>
 <par endsync="first">
  <audio src="Sound.wav"/>
  <img src="Vertgo.png" region="img_grand"/>
 </par>
 <par id="page2">
  <text src="Vertigo-Title.rt" region="txt1"/>
  <text src="Vertigo-Info.rt" region="txt2"/>
  <seq>
   <img src="Vertigo1.png"
          region="img_petit" dur="5s"/>
   <img src="Vertigo2.png"
        region="img_petit" dur="5s"/>
  </seq></par></seq>
</body>
```

Démo !

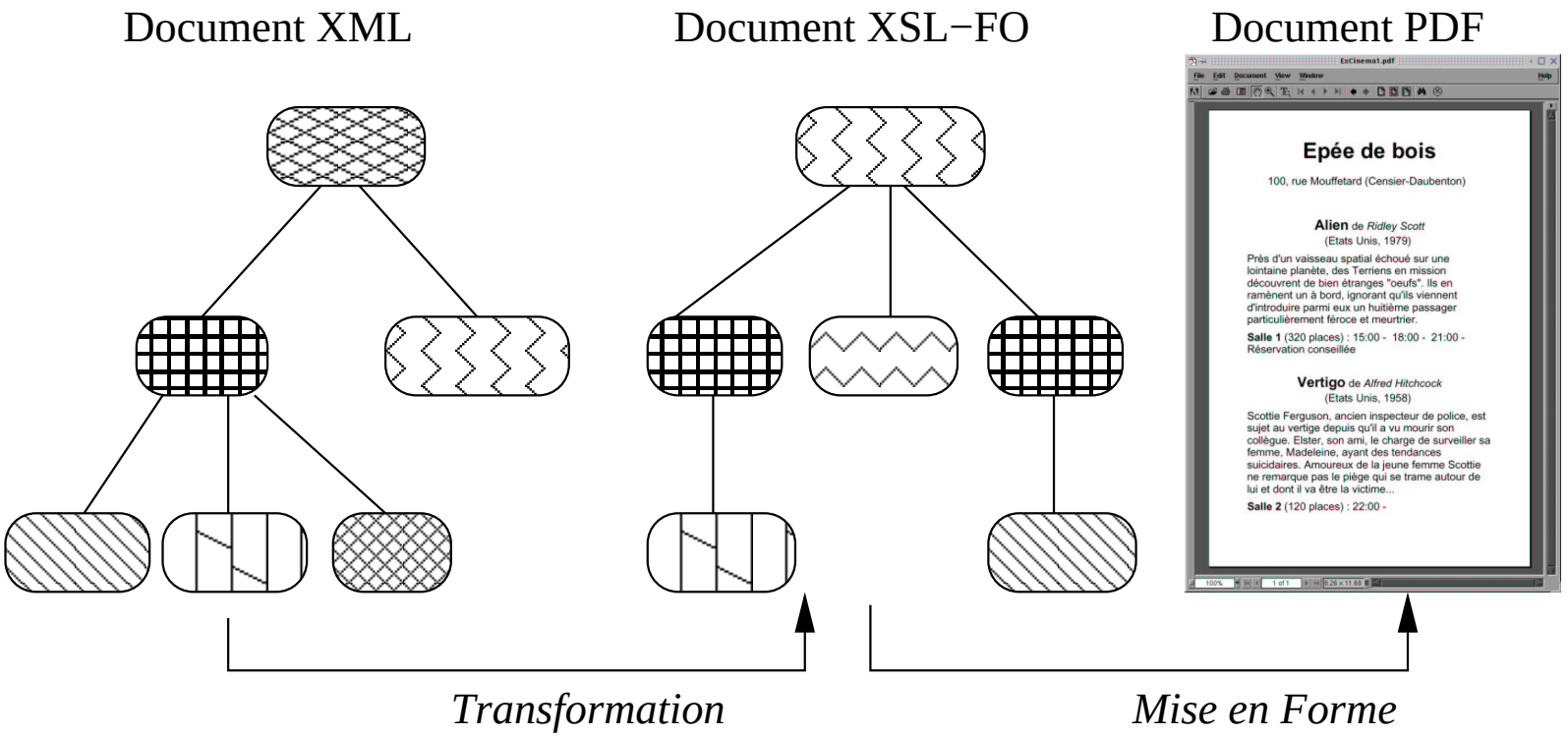
XSL-FO

Langage de **description** de documents avec XML.

- On indique les paramètres de mise en page (marges, taille des polices...)
- On place le contenu entre des balises de formatage

=> un processeur se charge de produire le document

Transformation, et mise en forme



Exemples d'un document XSL-FO

```
<?xml version="1.0" encoding="iso-8859-1"?>
<fo:root>
  <fo:layout-master-set>
    <fo:simple-page-master master-name="page"
      page-height="29.7cm" page-width="21cm"/>
  </fo:layout-master-set>
  <fo:page-sequence master-name='simple'>
    <fo:flow font-size="20pt">
      <fo:block>
        Ceci est le premier paragraphe,
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Démo ! Le programme de l'Epée de Bois!

L'approche XSL-FO

Traitement de texte WYSIWYG :

- On indique le contenu **et** la mise en forme
- Pbs :
  - Pas facile d'être expert en contenu **et** en mise en forme
  - Pas commode de penser aux deux à la fois

=> très difficile de faire de beaux documents (et impossible d'intégrer des contenus hétérogènes)

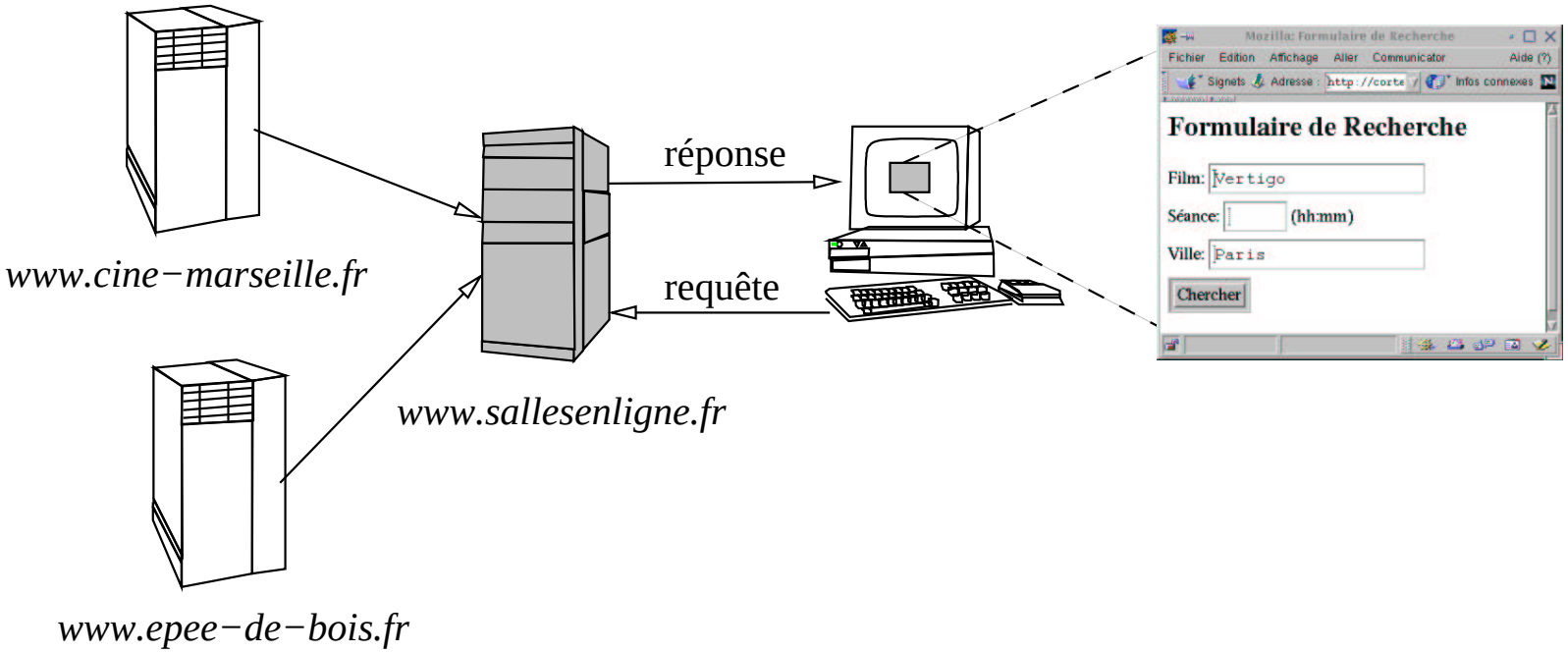
Quelques principes originaux

Avec XSL-FO :

- Un responsable pour le contenu (XML)
  - provenant de n'importe où (BD, sites, ...)
- Un responsable pour la mise en forme (XSL-FO)
  - décide de la présentation
- Un processeur pour produire le résultat

=> pas commode à apprendre ...

Exemple : moteur de recherche



Quelques idées

J’ai **mes** données

- Je leur ai défini une représentation
- Je leur applique des traitements (publication ou autre)
- **Je peux les transmettre** à quelqu’un d’autre (tout ou partie)

=> un **service** externe m’apporte une valeur ajoutée

Quel format ?

Mon problème :

- J’ai décrit mes données avec **mon** langage XML
- L’application attend des données dans **son** langage

Il faut :

- Décrire formellement les deux langages
- Faire une **traduction** de l’un à l’autre

Les DTD

Document Type Definition

- Pour définir la **structure** d'une classe de documents (d'un langage)
- Exemple : un élément de type texte :

```
<!ELEMENT TITRE ( #PCDATA ) >
```

- Exemple : un élément constitué d'une liste

```
<!ELEMENT FILM (TITRE, CINEMA, VILLE, URL?,
HEURE+)>
```

La DTD du moteur de recherche

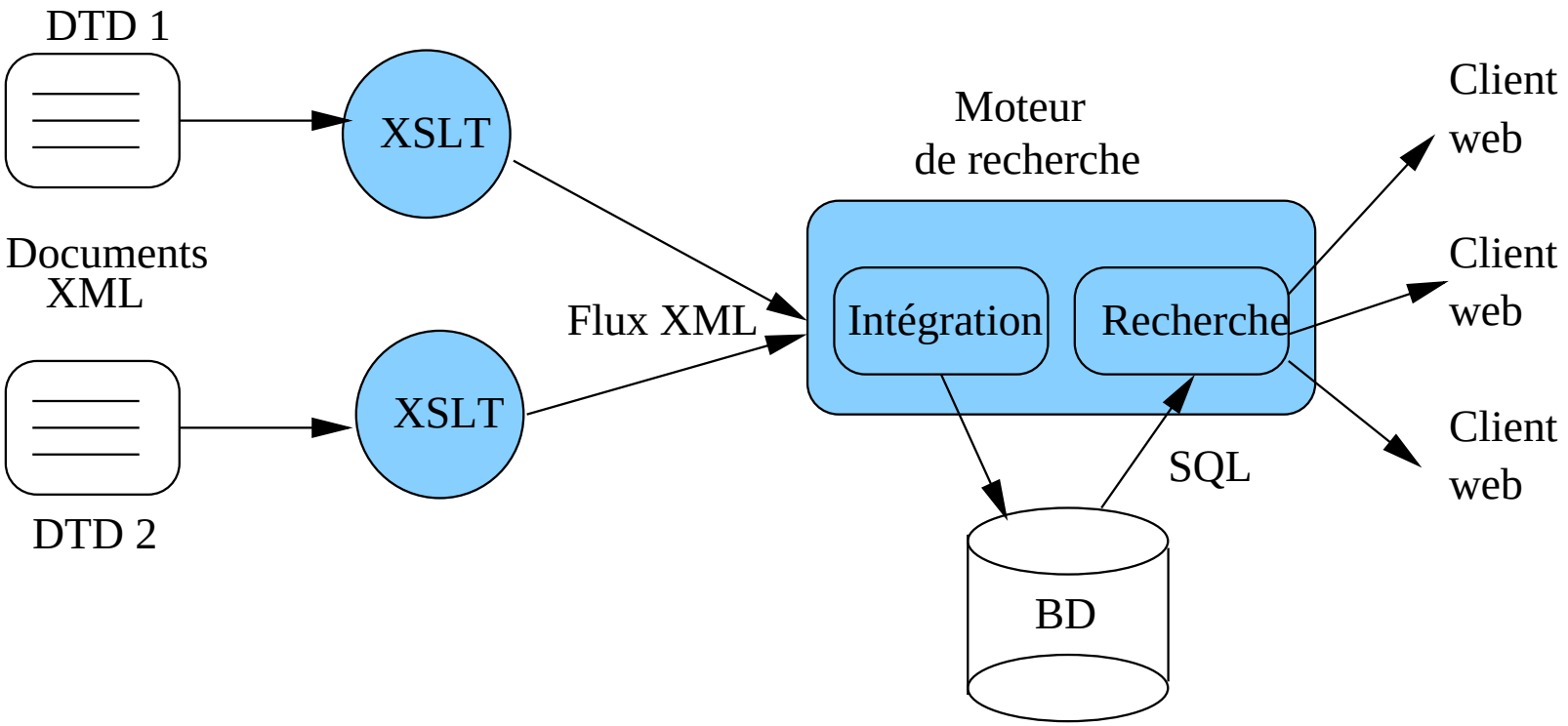
Un fichier auquel on peut faire référence dans un document :

```
1  <!ELEMENT FILM    ( TITRE, CINEMA, VILLE, URL?, HEURE+ ) >
2  <!ELEMENT TITRE   ( #PCDATA ) >
3  <!ELEMENT CINEMA  ( #PCDATA ) >
4  <!ELEMENT VILLE   ( #PCDATA ) >
5  <!ELEMENT URL     ( #PCDATA ) >
6  <!ELEMENT HEURE   ( #PCDATA ) >
```

Document valide : conforme à une DTD.

Architecture

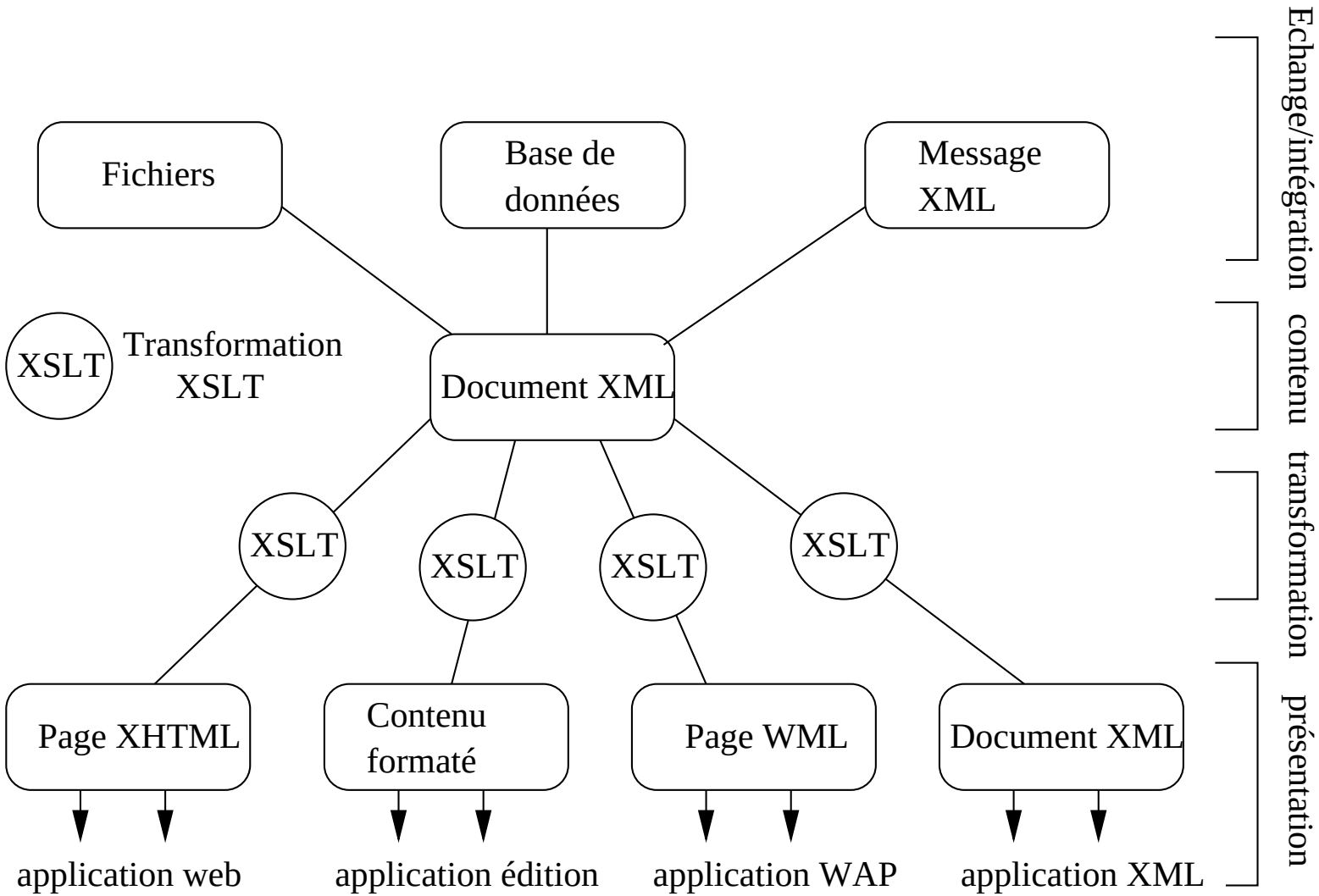
(Démo)



Le document intégrateur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MOTEUR [
 <!ENTITY EpeeDeBois
     SYSTEM "http://epee-de-bois.fr/EDB.xml">
 <!ENTITY CineMarseille
     SYSTEM "http://cine-marseille.fr/CM.xml">
]>
<MOTEUR>
  <CINEMA>
    &EpeeDeBois;
   </CINEMA>
  <CINEMA>
    &CineMarseille;
   </CINEMA>
</MOTEUR>
```

Gestion de l'information avec XML



Récapitulons !

XML = format d'échange de données entre application

- Permet de définir des « langages » pour décrire des données (« méta-langage »)
- De nombreux outils d'analyse, *parsing*, interrogation, ...
- Transformation d'un langage à un autre avec XSLT

=> Bien adapté au web.