



Echange et publication d'information sur l'Internet

Une étude de cas avec XML

Ce qu'on ne sait pas faire avec le Web

Le principal langage du Web est HTML

- ⑥ HTML est un langage pour **présenter** des informations à **l'écran**.
 - △ il ne permet pas d'échanger des données
 - △ il ne permet pas un traitement des données autre que l'affichage

On ne sait pas interpréter des données fournies en HTML

Nouvelles utilisations du Web

Quelques applications :

- **Commerce électronique** : les entreprises veulent échanger des informations (pas pour les afficher !)
- **Moteur de recherche** : si je sais interpréter les données transmises, je peux les indexer efficacement
- **Services en ligne** : je peux envoyer mes données à un serveur pour leur appliquer un traitement donné (ex : publication)

Nouveaux besoins

Mots clés : **échange** et **publication** de données.

- **Le réseau est hétérogène** : les données doivent être représentées indépendamment d'une machine donnée.
- **Les applications sont variées** : les données doivent être représentées indépendamment d'une application.
- **Une application = un format de données** : il faut pouvoir transformer facilement les données d'un format à un autre.

Pourquoi XML ?

Le langage XML est une réponse à ces besoins :

- **Un document XML est au format ASCII** : il voyage (relativement) facilement
- **XML n'est pas lié à un mode d'utilisation** : chacun peut se définir son propre « langage », (SMIL, SVG, WML, etc)
- **Beaucoup d'outils de manipulation** : DOM, SAX, XSLT, XPath, XQuery, etc

Une étude de cas XML

Une présentation, basée sur une étude de cas, des apports de XML :

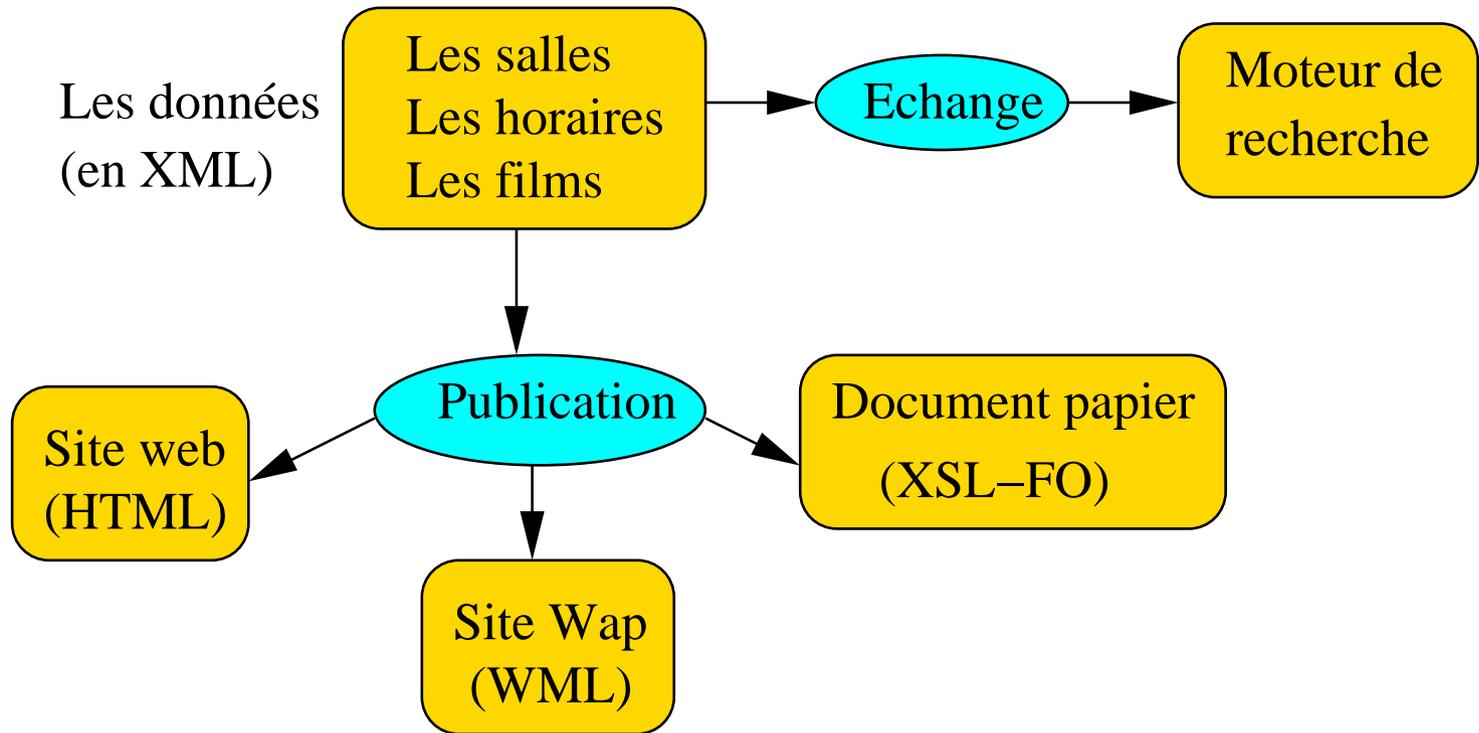
- XML, format universel
- Publication avec XSLT
- Échange et intégration de données
- XML et bases de données

Les questions : XML, c'est quoi, et pour quoi faire

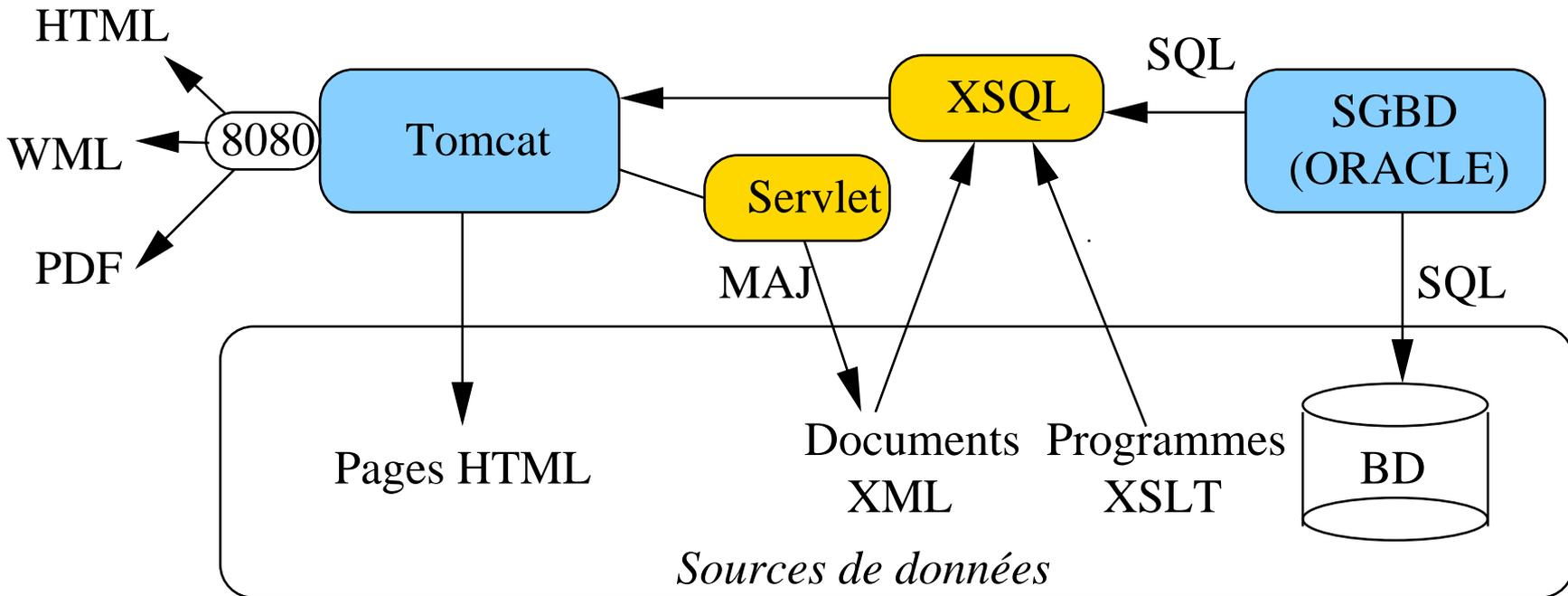
L'Officiel des spectacles !

- 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

Echange et publication des données



L'architecture du site



Le programme de L'Epée de Bois

- sur son site web
- sur son site WAP
- sous forme tract PDF
- dans le moteur de recherche SallesEnLigne.com

L'information ?

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

Les problèmes ?

Je voudrais :

- ... éviter de gérer n fois le programme de mon cinéma, sous n formats différents (BD, fichier, traitement de texte, tableur)...
- ... rendre mon programme disponible sous plusieurs formes (sans trop d'effort...)
- ... **récupérer** l'information existante chez d'autres
- ... **donner** mes informations à ceux qui peuvent lui apporter une valeur ajoutée

Les solutions XML ?

■ **Format universel :**

représentation la plus simple possible (chaîne de caractères) d'un contenu
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>
```

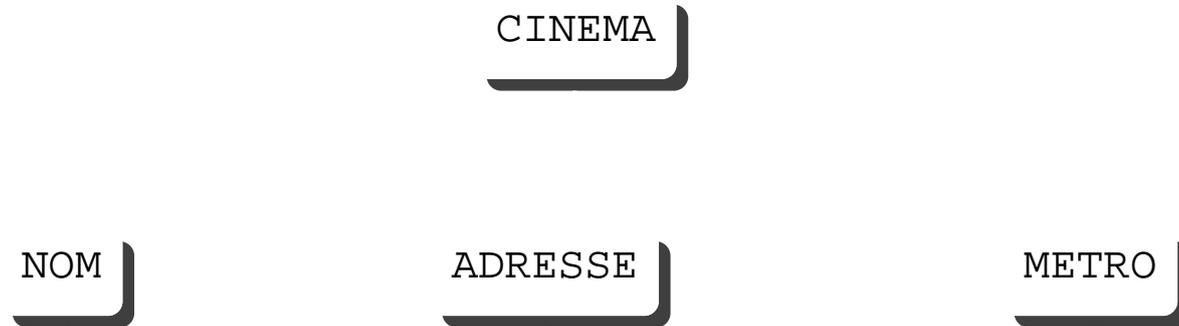
Le même, mieux présenté

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



Épée de bois

100, rue Mouffetard

Censier-Daubenton

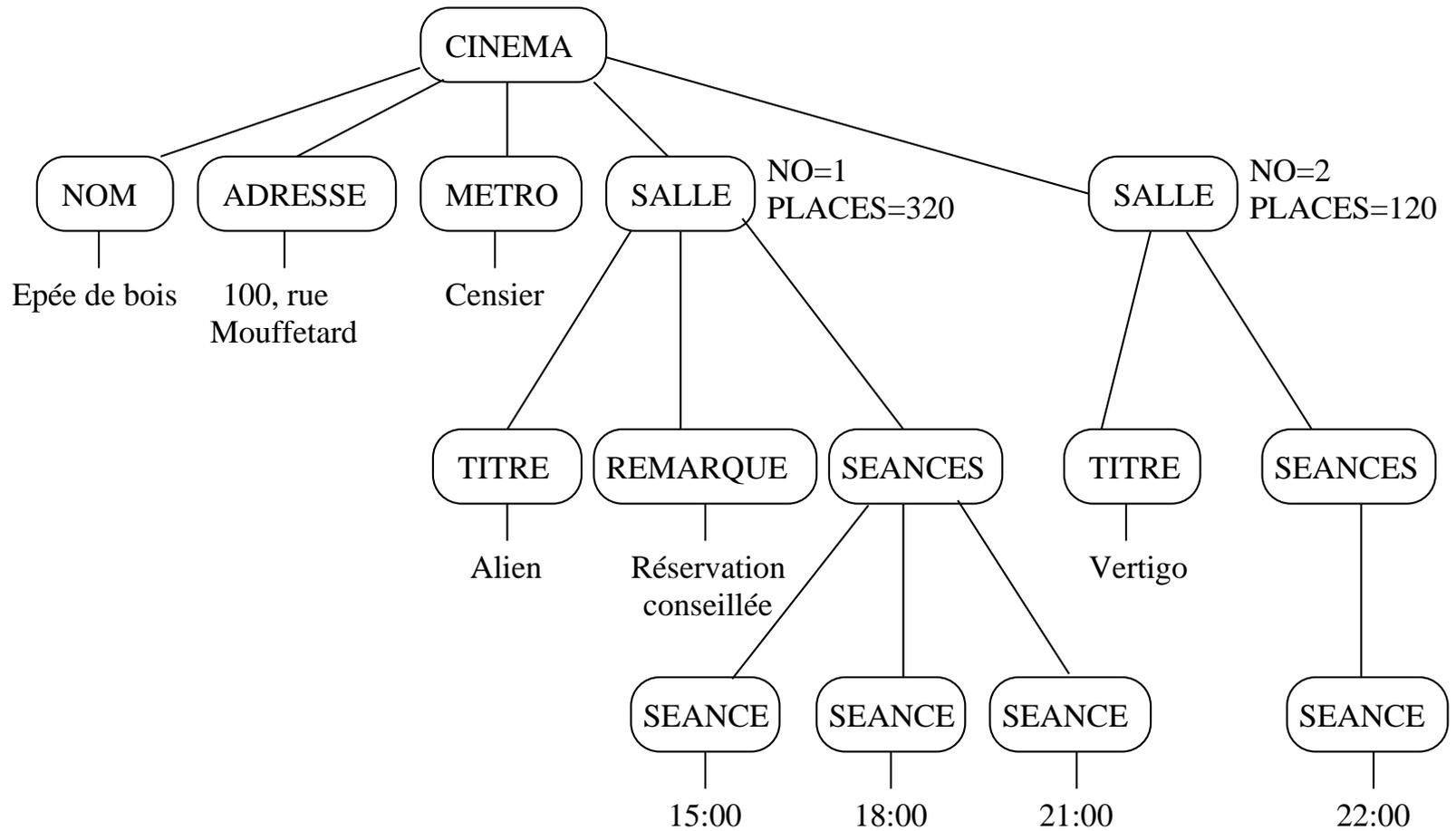
Traiter un document XML = extraire des informations d'un arbre.

Un exemple plus complet, avec attributs

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

  <SALLE NO='1' PLACES='320'>
    <TITRE>Alien</TITRE>
    <REMARQUE>Reservation conseillée</REMARQUE>
    <SEANCES>
      <SEANCE>15:00</SEANCE>
      <SEANCE>18:00</SEANCE>
      <SEANCE>21:00</SEANCE>
    </SEANCES>
  </SALLE>
</CINEMA>
```

Sous forme d'arbre



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

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

Indépendant de la **représentation physique**

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

Origine d'un document XML

XML permet **d'intégrer** des contenus provenant d'origines diverses

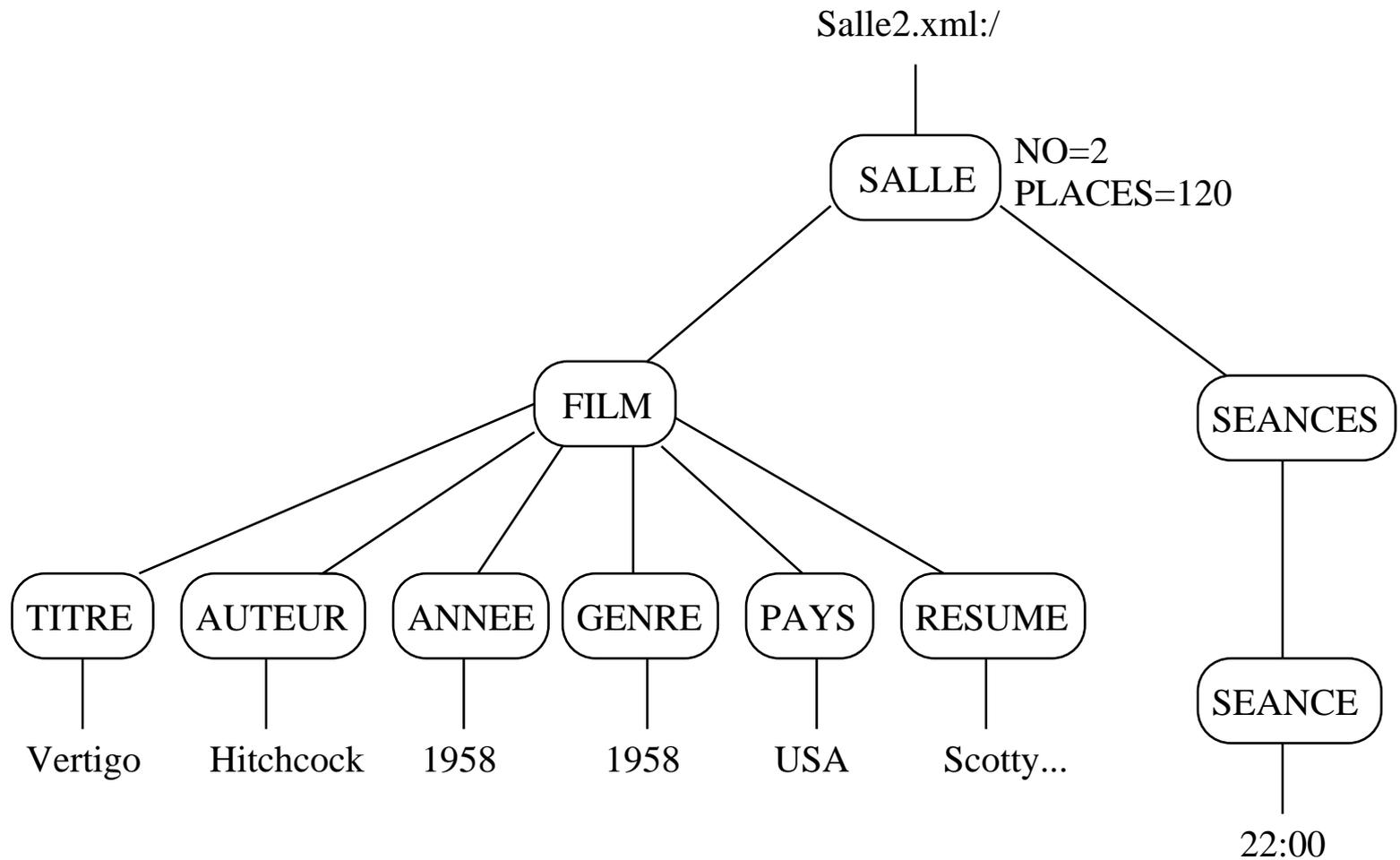
- D'un traitement de texte
- d'un site web
- d'une base de données
- d'un fichier...

Encore la même idée : rendre le contenu indépendant de l'application

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>
```

Premier bilan

Disposer d'une et une seule représentation

- J'utilise un traitement de texte ? je suis prisonnier du format
- Je stocke dans une base de données ? idem

Créer un langage pour décrire nos données

- Ne pas utiliser HTML, dédié à la présentation dans un navigateur
- => utiliser XML, et convertir vers HTML

Publication de données avec XSLT

XSLT : pour quoi faire

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

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

Application

Nous avons décrit notre cinéma avec notre propre langage. XSLT va permettre de traduire ce langage vers 3 langages XML spécialisés :

- HTML, pour la présentation de documents web
- WML pour la présentation de documents WAP
- XSL-FO pour la production de documents papier

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 (d mo)

```
<html>
  <head><title>Film: Vertigo</title></head>
  <body bgcolor="white">
    <p><img SRC="Vertigo.png">
      <h1><i>Vertigo</i></h1>
      Drame, <i>Etats Unis</i>, 1958
    </p>
    <p>
      Mis en sc ne par <b>Alfred Hitchcock</b>
      <h3>R sum </h3>Scottie Ferguson,
      ancien inspecteur de police, est sujet
      au vertige depuis qu'il a vu mourir son
      coll gue...
    </p>
  </body>
</html>
```


Le rôle de XSLT

XSLT doit nous permettre :

- 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 (``, `<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>
```

Création d'un site WAP

On envoie un **ensemble** de cartes :

- Dotées d'une **identité** :

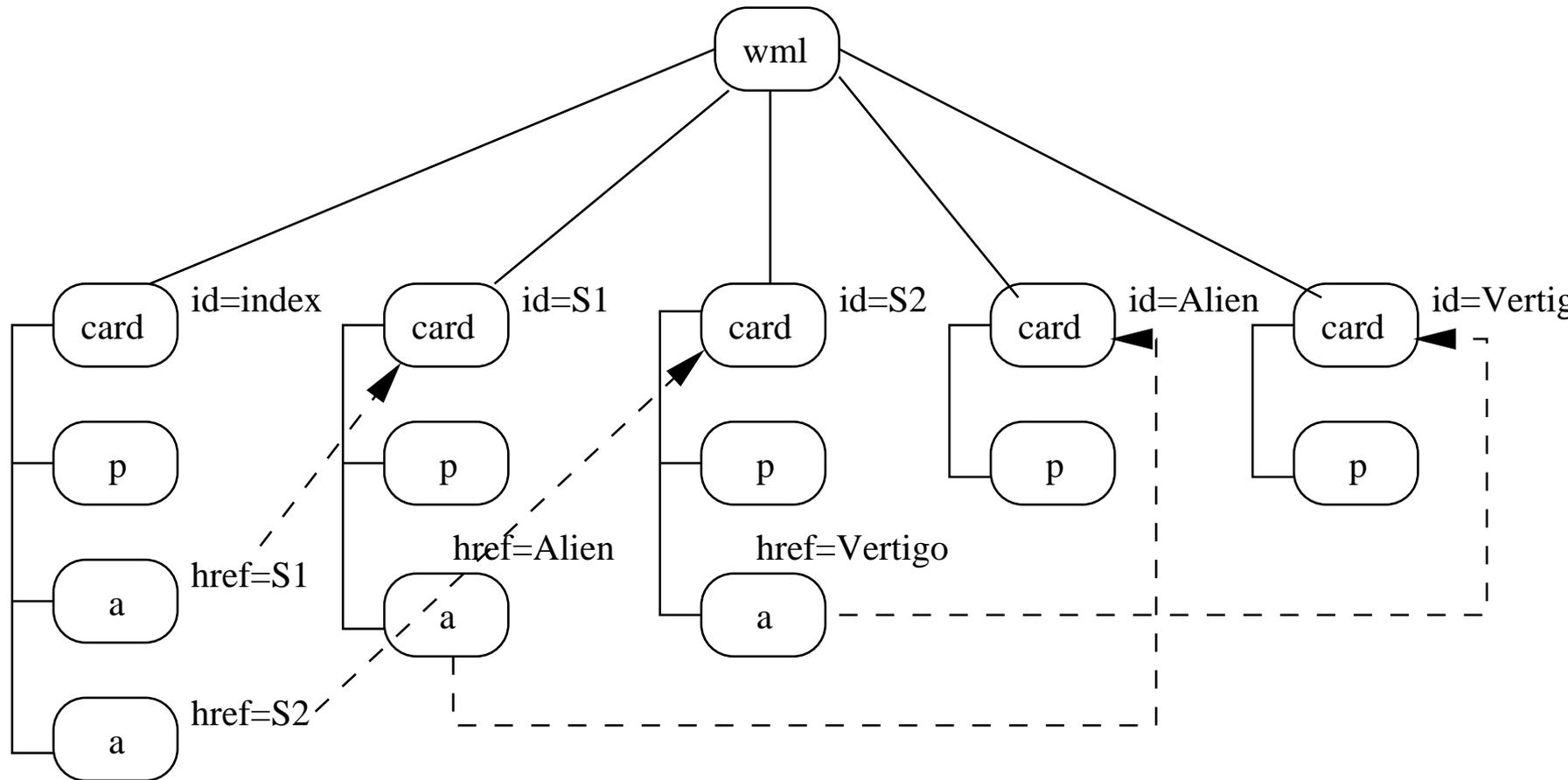
```
<card id="Alien"> ... suite de la carte  
</card>
```

- **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



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>
```

Version XSL-FO

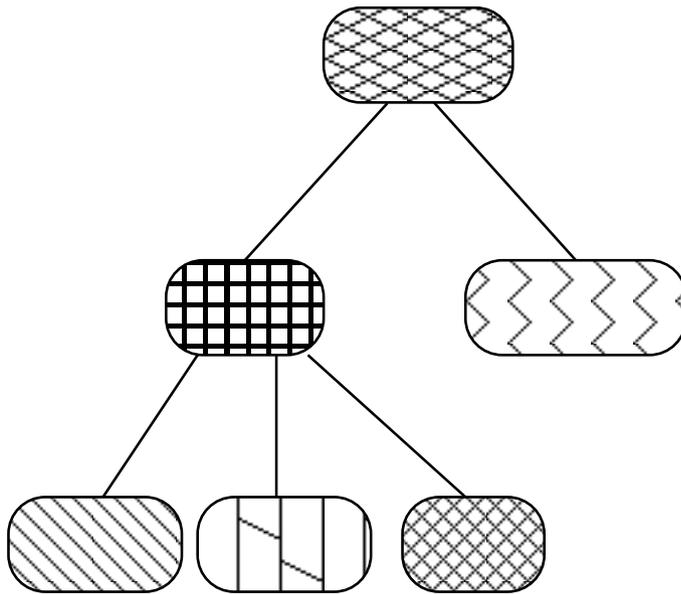
Langage de **description** de documents avec XML.

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

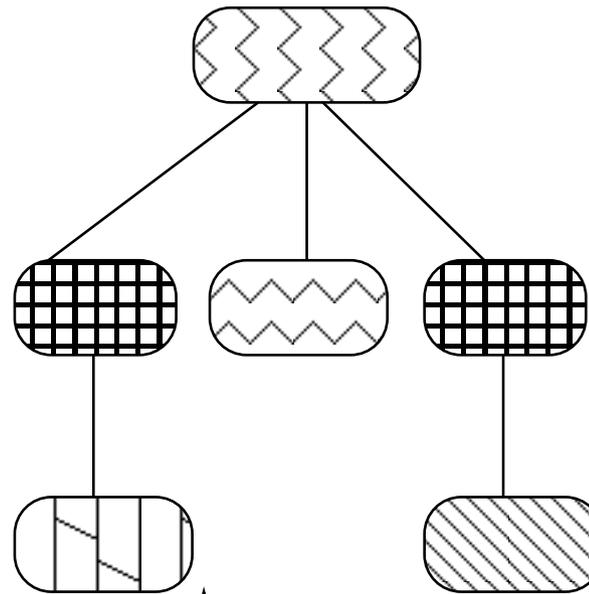
=> un processeur se charge de produire le document

Transformation, et mise en forme

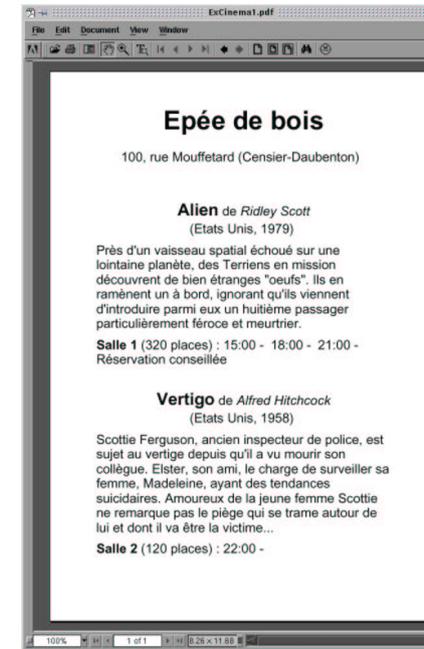
Document XML



Document XSL-FO



Document PDF



Transformation

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 !

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

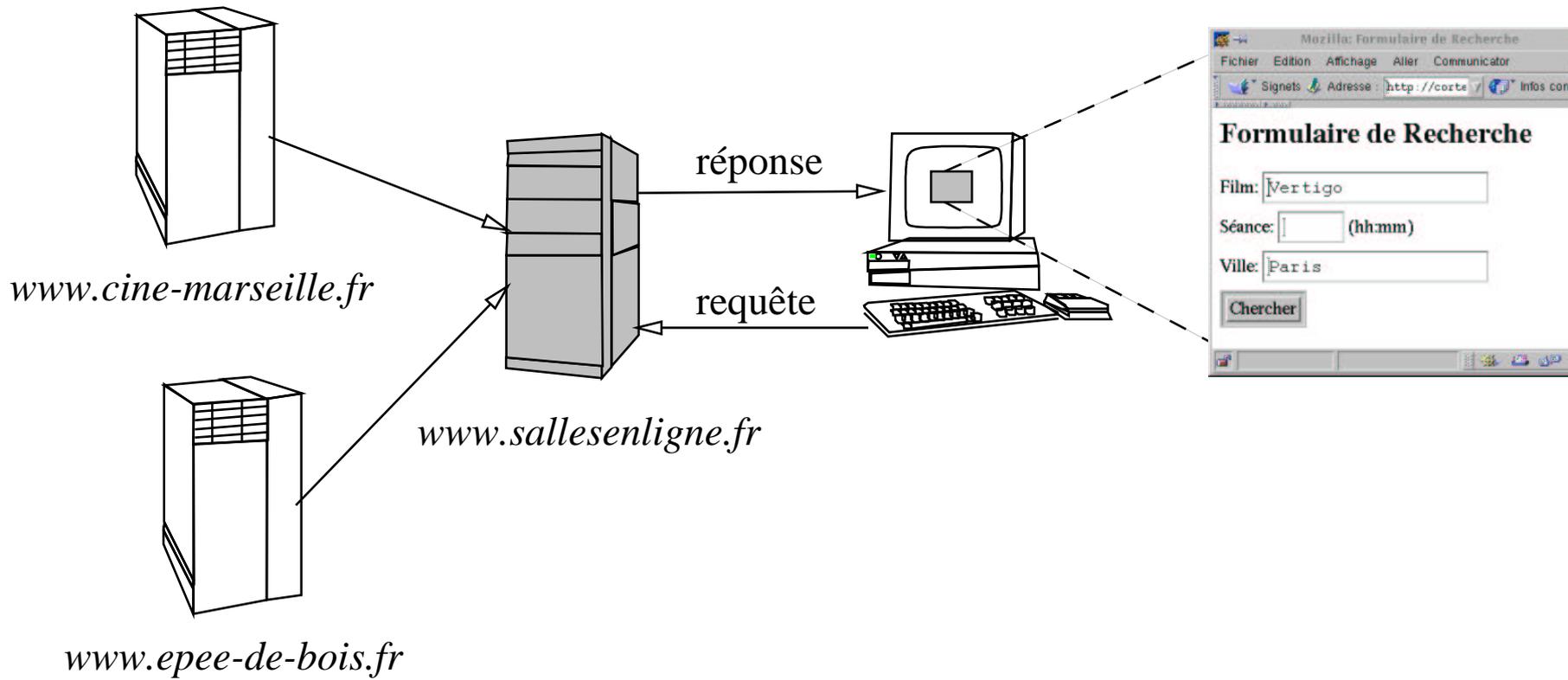
Avec XSL-FO :

- 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 ...

Échanges et intégration de données

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

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+ ) >
```

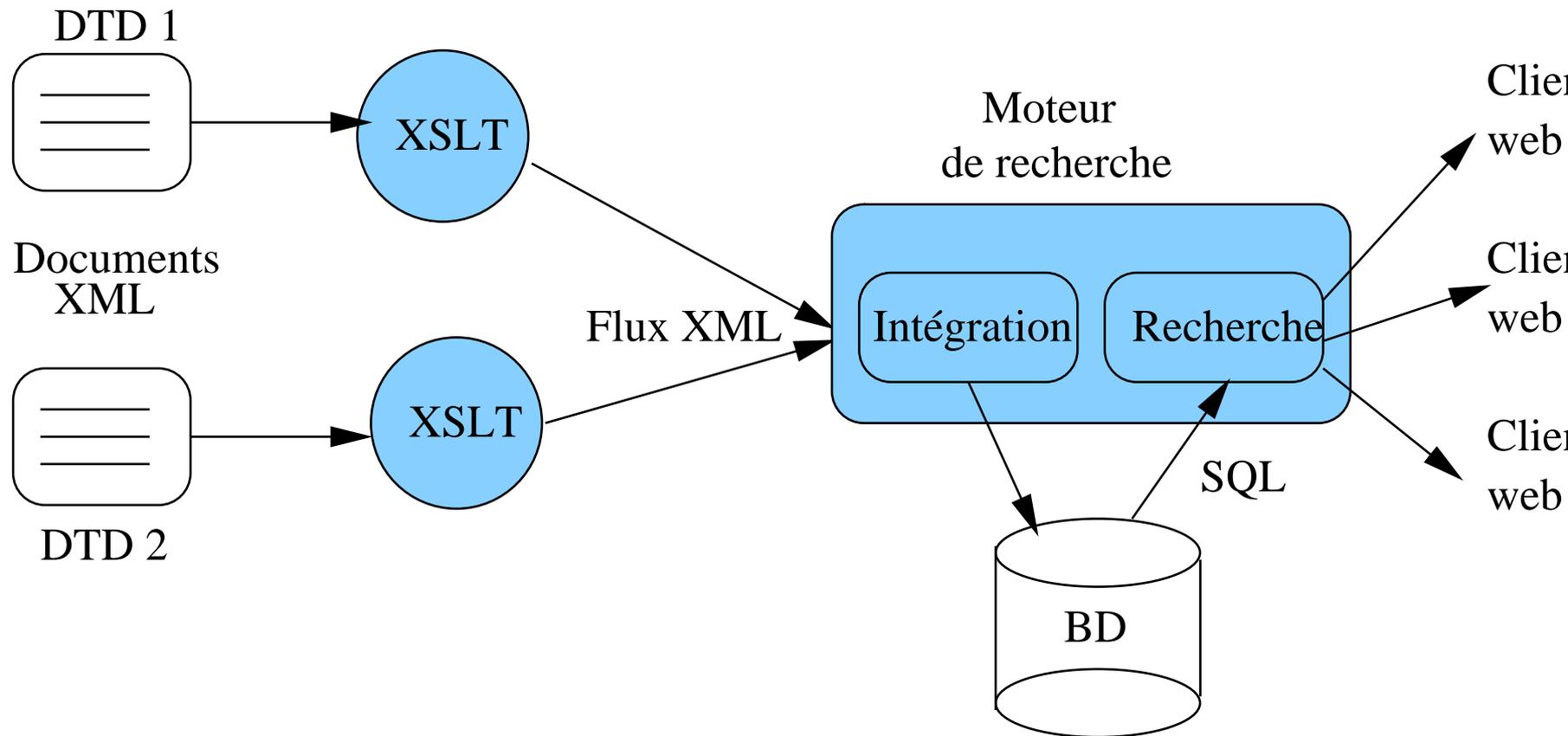
La DTD du moteur de recherche

Un fichier auquel on peut faire référence dans un document :

```
<!ELEMENT FILM      ( TITRE, CINEMA, VILLE, URL ) >
<!ELEMENT TITRE     ( #PCDATA ) >
<!ELEMENT CINEMA    ( #PCDATA ) >
<!ELEMENT VILLE     ( #PCDATA ) >
<!ELEMENT URL       ( #PCDATA ) >
<!ELEMENT HEURE     ( #PCDATA ) >
```

Document valide : conforme à une DTD.

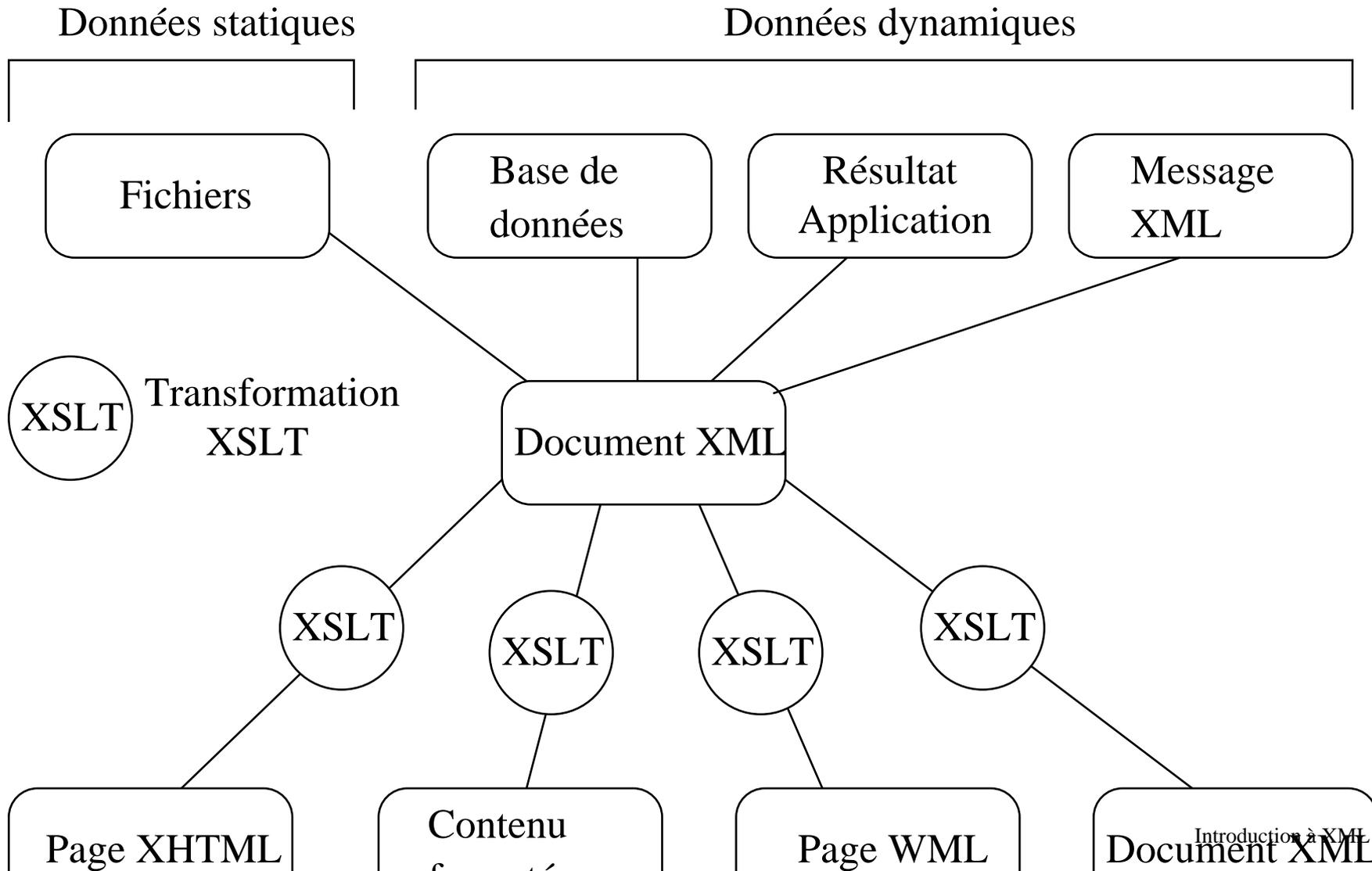
Architecture (Démo)



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.

Ce qu'on apprend dans le cours

À l'issue des cours+TD+projet, vous devriez savoir :

- Comment créer un (ou des) document(s) XML corrects pour structurer de l'information, textuelle ou issue d'une BD.
- Analyser ou mettre à jour ces documents avec les API DOM, SAX et le langage XPath.
- Effectuer des transformations avec XSLT.