XML : documents et outils

Pierre-Alain Reynier
Université de Provence

pierre-alain.reynier@lif.univ-mrs.fr
http://www.lif.univ-mrs.fr/~preynier/XML

Cours adapté du travail de
Rémi Eyraud, Silvano Dal Zilio...

Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

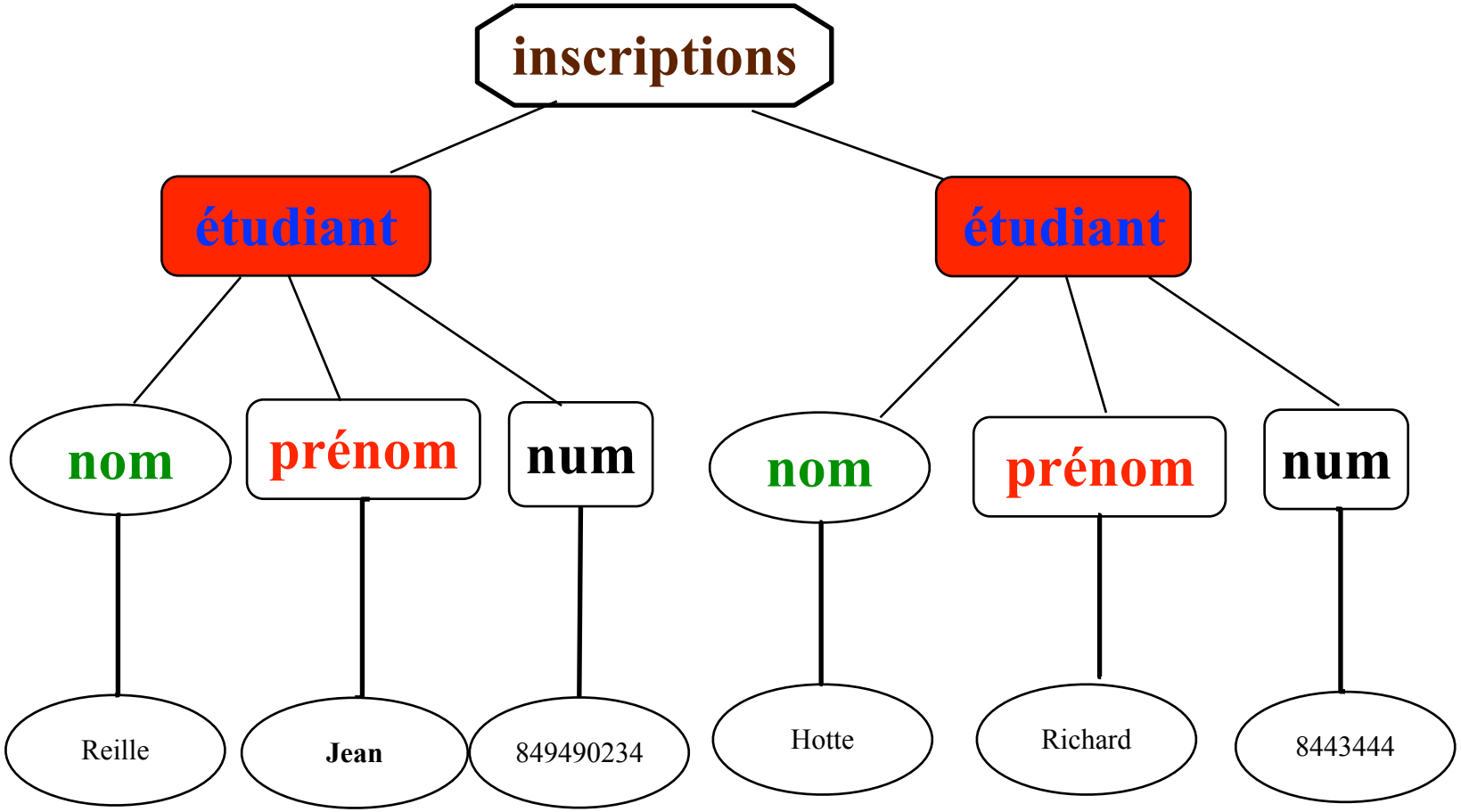
Qu'est-ce que le XML ?

- The eXtensible Markup Language (XML)
- Format universel pour représenter les données (semi-) structurées et les documents
- Format textuel
- Langage de balises (markup language), comme HTML, à la différence que les balises n'ont pas de signification a priori.

Un exemple

```
<cours>
    <nom>XML</nom>
    <université>UP</université>
    <inscriptions>
        <étudiant>
            <nom>Reille</nom>
            <prénom>Jean</prénom>
            <num>849490234</num>
        </étudiant>
        <étudiant>
            <nom>Hotte</nom>
            <prénom>Richard</prénom>
            <num>8443444</num>
        </étudiant>
    </inscriptions>
</cours>
```

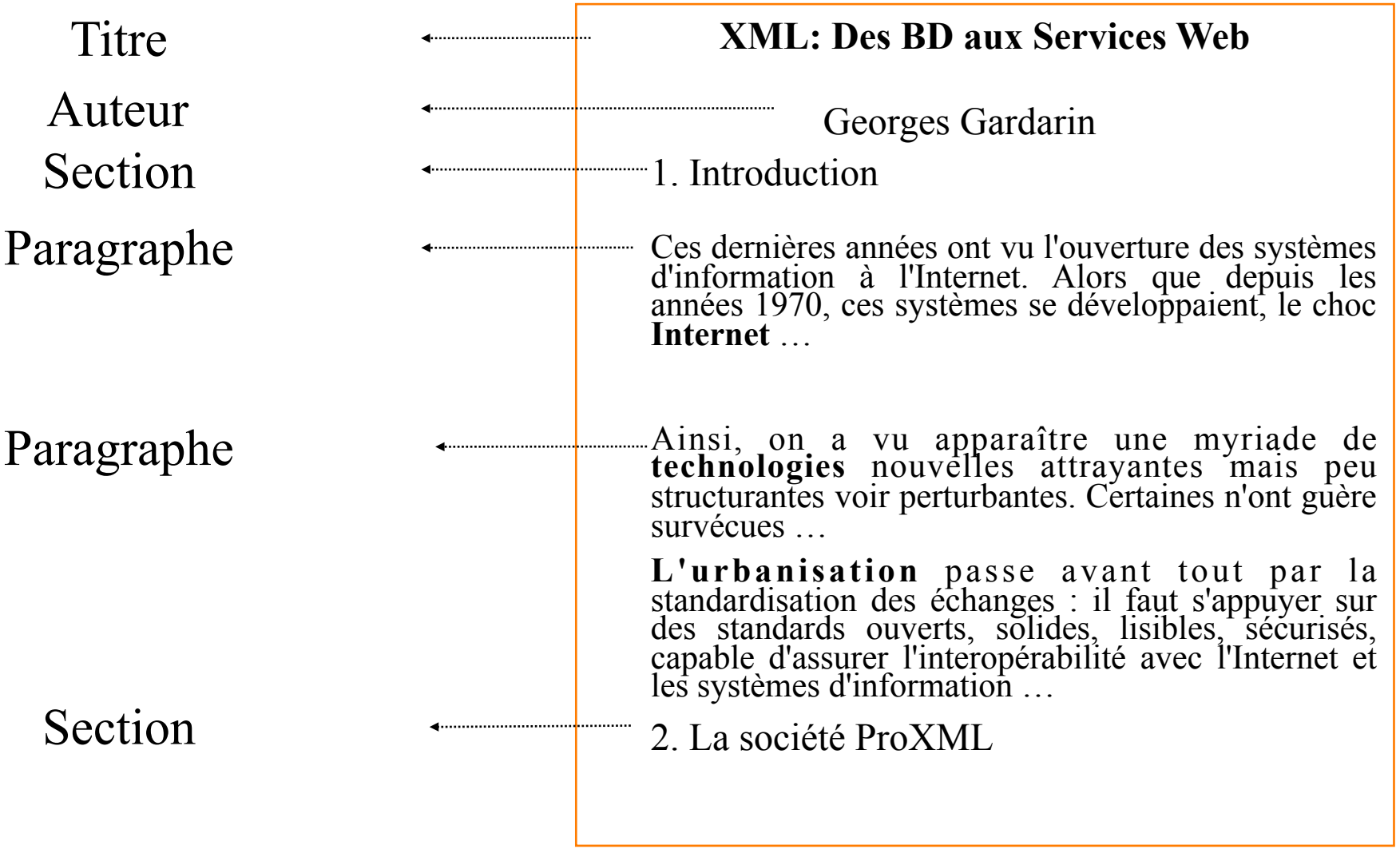
C'est un arbre (orienté) !



C'est quoi XML (la suite) ?

- Séparation du fond de la forme.
  - Forme = présentation à partir de la structure (style)
  - Fond = structure + données (contenu)

Fond et forme



Fond et forme : vue balisée

```
<Livre>
    <Titre> XML : Des BD aux Services Web</Titre>
    <Auteur>Georges Gardarin</Auteur>
    <Section titre = "Introduction">
        <Paragraphe> Ces dernières années ont vu l'ouverture des systèmes d'information à
            l'Internet. Alors que depuis les années 1970, ces systèmes se développaient, le choc
            Internet …
        </Paragraphe>
        <Paragraphe> Ainsi, on a vu apparaître une myriade de technologies nouvelles
            attrayantes mais peu structurantes voir perturbantes. Certaines n'ont guère
            survécues …
        </Paragraphe>
        <Paragraphe> L'urbanisation passe avant tout par la standardisation des échanges :
            il faut s'appuyer sur des standards ouverts, solides, lisibles, sécurisés, capable
            d'assurer l'interopérabilité avec l'Internet et les systèmes d'information …
        </Paragraphe>
    </Section>
    <Section titre= "La Société ProXML"> …
    </Section>
</Livre>
```

Les balises (tags) peuvent porter plus ou moins de sémantique

C'est quoi XML (fin) ?

- Libre de droits, indépendant des plates-formes et correctement pris en charge.
- Format self descriptif !
- XML utilise la notion de DTD (Document Type Definition) pour décrire des contraintes sur les données

Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

Les origines du XML, SGML et HTML

- 1969 C. Goldfarb, E. Mosher, R. Lorie inventent GML chez IBM
- GML a été créé pour éditer des documents, les mettre en page et les partager au sein de systèmes de gestion éditoriaux
- 1978 Goldfarb prend la tête d'un comité « Computer Language for the Processing of Text » au sein de l'American National Standards Institute (ANSI).

Qu'est-ce que SGML ?

- Une norme internationale :
  - Standard Generalized Markup Language
  - ISO 8879 - 1989
- Un métalangage de balisage de documents
  - lisible par l'être humain et traitable par une machine
  - permet de définir des langages de balisage
- Les documents sont balisés conformément à la grammaire (la DTD)
  - instances de DTD
  - permet un balisage sémantique du fond.
- Implique la notion de validité d'un document

SGML : objectifs

- Séparation du fond et de la forme
  - possibilité de multiples présentations
  - un seul document en SGML
  - plusieurs formats : Postcript, HTML, etc.
- Support de traitements sur le contenu des documents sans prise en compte de la forme
- Proposition d'un cadre défini pour l'expression des modèles documentaires (validité, contrôle)
- Intégration d'un format de stockage et d'échange normalisé

SGML : critiques

- Très lourd et complexe pour la mise en œuvre de documents respectant ce format
- Une grande rigueur est demandée à l'entrée des documents
- Standard complexe et complet pour le traitement des documents
- Liens hypertextes possibles mais complexes

HTML : présentation

- Proposé par le W3C comme format de documents sur le Web.
- Langage simple avec des balises standardisées permettant la mise en forme d'un texte.
- Standard reconnu par tous les navigateurs.
- Langage très populaire sur le Web

```
<HTML>
   <HEAD>
      <TITLE> Exemple </TITLE>
   </HEAD>
   <BODY>
      <H1>Contenu du document</H1>
      <A HREF = " http://www.server.fr/Info/dir/test.html " >
          une référence externe
      </A>
   </BODY>
</HTML>
```

HTML : inconvénients

- Normalisation des différentes balises difficile :
  - les constructeurs ont eu tendance à définir leurs propres balises pour répondre à leurs besoins (incompatibilité)
  - HTML 4.0
    - boutons, tables, applets, objects, graphiques, maths, ...
    - styles, frames, protections, ...
- Mises à jour difficiles :
  - données utiles et mises en forme ;
  - restructuration ou remise en forme de l'ensemble des pages du site fastidieux.
- Mélange le fond et la forme
  - méta-données avec la présentation
  - Pages conçues pour un seul type de terminal (réécriture nécessaire pour les versions mobiles)

Feuilles de style

- Introduites pour diversifier les présentations
- CSS (*Cascading Style Sheet*)
  - mécanisme d'héritage entre nœuds
  - une balise hérite de la parente
  - seulement ce qu'elle spécifie est redéfini
- Recommandation W3C en décembre 1996
- Mécanisme simple pour ajouter un style aux documents Web
  - fonte, taille, couleur, etc...
- Utilisables avec XML

Exemple de CSS

```
<LINK REL="stylesheet" HREF="fichier.css">
```

```
@import "truc.css"
BODY {
    color: #000 ;
    background: #FBFBFF ;
    margin-left: 9% ;
    margin-right: 6% ;
    font-family: "Helvetica", sans-serif ;
    line-height: 1.35 ;
}
```

```
[...]

TD, TH {
    font-family: “Helvetica”,
        sans-serif ;
    line-height: 1.35 ;
}

H1, H2 {
    margin-top: 1.2em ;
    margin-left: -7%;
    color: # 900;
    clear: both;
}

[...]
```

SGML et HTML : Résumé

- SGML
  - langage de la GED plutôt complexe
  - très utilisé dans l'industrie
- HTML
  - spécialisation de SGML
  - adapté à la présentation sur Internet
  - inadapté à l'échange entre programmes
- **XML :**
  - plus simple que SGML
  - plus ouvert que HTML
  - Recommandation W3C du 10 février 1998

World Wide Web Consortium

- W3C - Fondé en 1994
- Consortium industriel international accueilli par différents sites
  - MIT/CSAIL aux Etats-Unis
  - ERCIM en France
  - Keio University au Japon
- environ 400 membres industriels ou académiques

W3C : dans quel but ?

- Accroître le potentiel du Web
- Développer des protocoles communs
- Assurer l'inter-opérabilité sur le Web entre les différents systèmes
- Stock d'informations sur les standards et les normes pour développeurs et utilisateurs
  - Code référence pour présenter et promouvoir les différents standards
  - Prototypes variés et exemples d'applications

Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

Quelques définitions

- Le XML est un « métalangage » permettant d'échanger de l'information, principalement sur le web.
- On peut définir son propre jeu de balises (jargon) et le sens qu'on leur donne : on appelle ça une application XML (document DTD, XML Schema ou Relax NG)
- Documents bien formés :
  - balises correctement imbriquées
  - analysable (parsable) et manipulable
- Le XML définit une grammaire stricte et relativement simple → outils génériques pour les documents bien formés
- Un document XML qui respecte les normes d'une application XML est valable
(un document valable est bien formé).

Quel est le rôle du XML

- XML standardise la manière dont l'information est :
  - échangée
  - présentée
  - archivée
  - retrouvée
  - transformée
  - cryptée
  - ...

La philosophie XML

“be strict in what you produce,
be generous in what you accept”

Exemple :

Vos applications doivent fonctionner même si vous vous attendiez à :

```
<commande>
   <produit>papier</produit>
   <quantite>2</quantite>
</commande>
```

Mais que vous recevez :

```
<commande>
   <produit>papier</produit>
   <quantite>2</quantite>
   <qualite>brouillon</qualite>
</commande>
```

A quoi sert le XML ?

- XML n'est pas limité à un type d'application (tant que ça reste textuel).
- Message Oriented Middleware (MOM) !
  - destiné aux échanges de données machine-machine
  - standard pour la structuration des données
  - exemple: B2B (Business-to-Business communication)
- Presentation Oriented Publishing (POP) !
  - destiné aux données qui seront “consommées” par des êtres humains!
  - utilisé par les butineurs (browsers) et les éditeurs

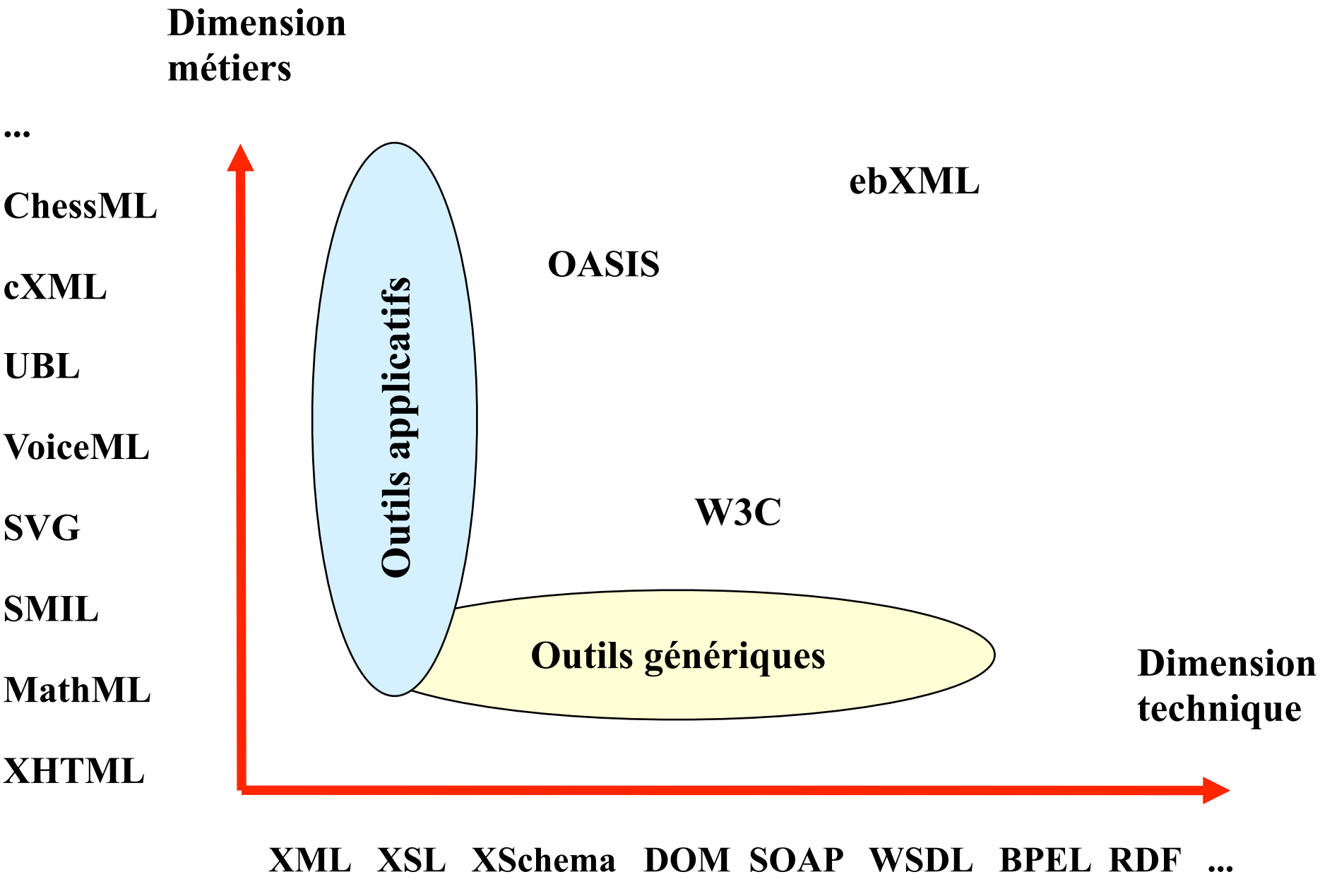
Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

Pourquoi XML?

- Définir vos propres langages d'échange
  - Commande, facture, bordereau de livraison, etc.
- Modéliser des documents et des messages
  - Modèle logique de données
  - Eléments typés agrégés (DTD, XML Schema)
  - Passerelle avec Unified Modelling Language (UML)
- Publier des informations
  - Neutre du point de vue format
  - Mise en forme avec des feuilles de style
- Archiver des données
  - Auto-description des archives

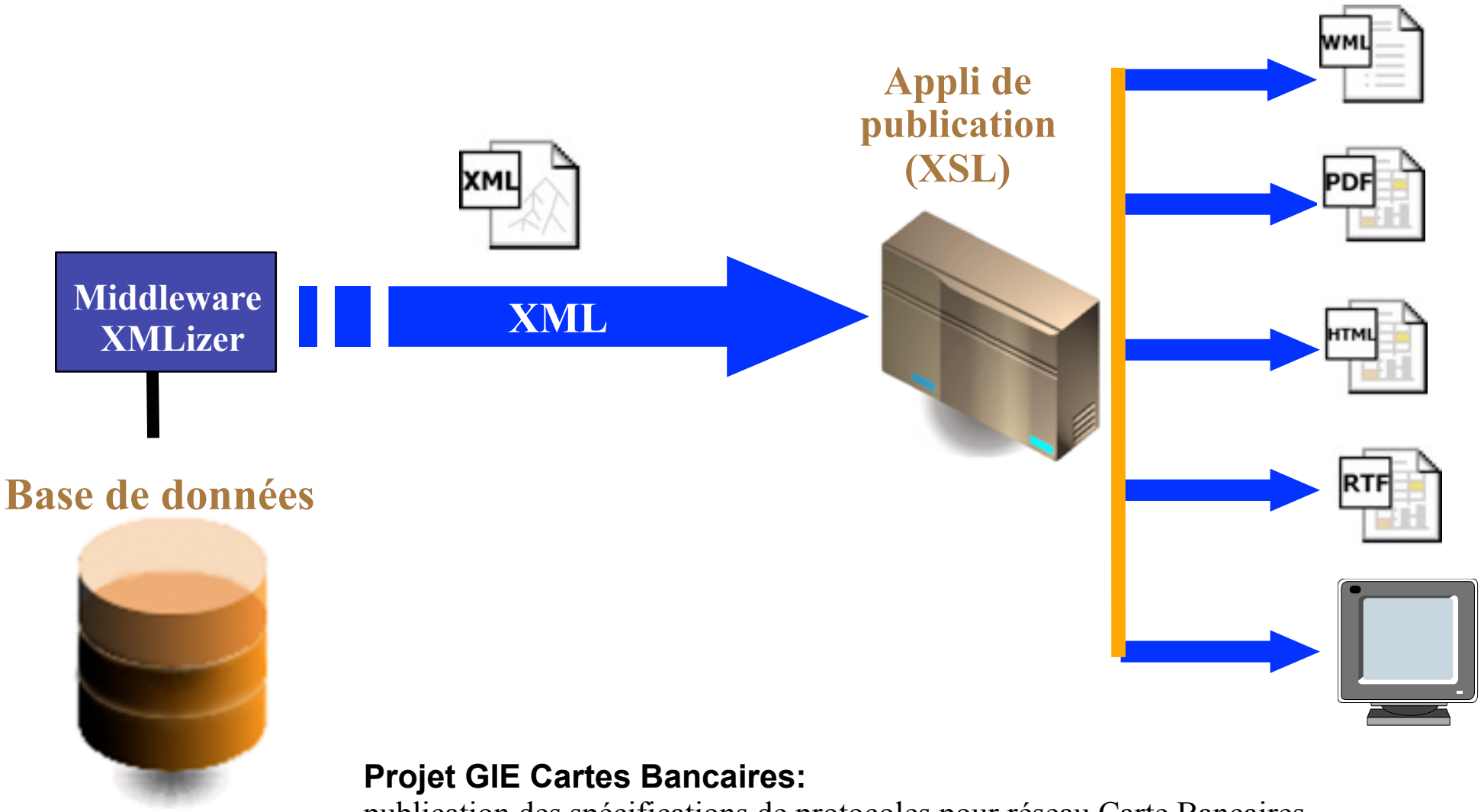
La galaxie de standards



"Lingua Franca" du 21e siècle

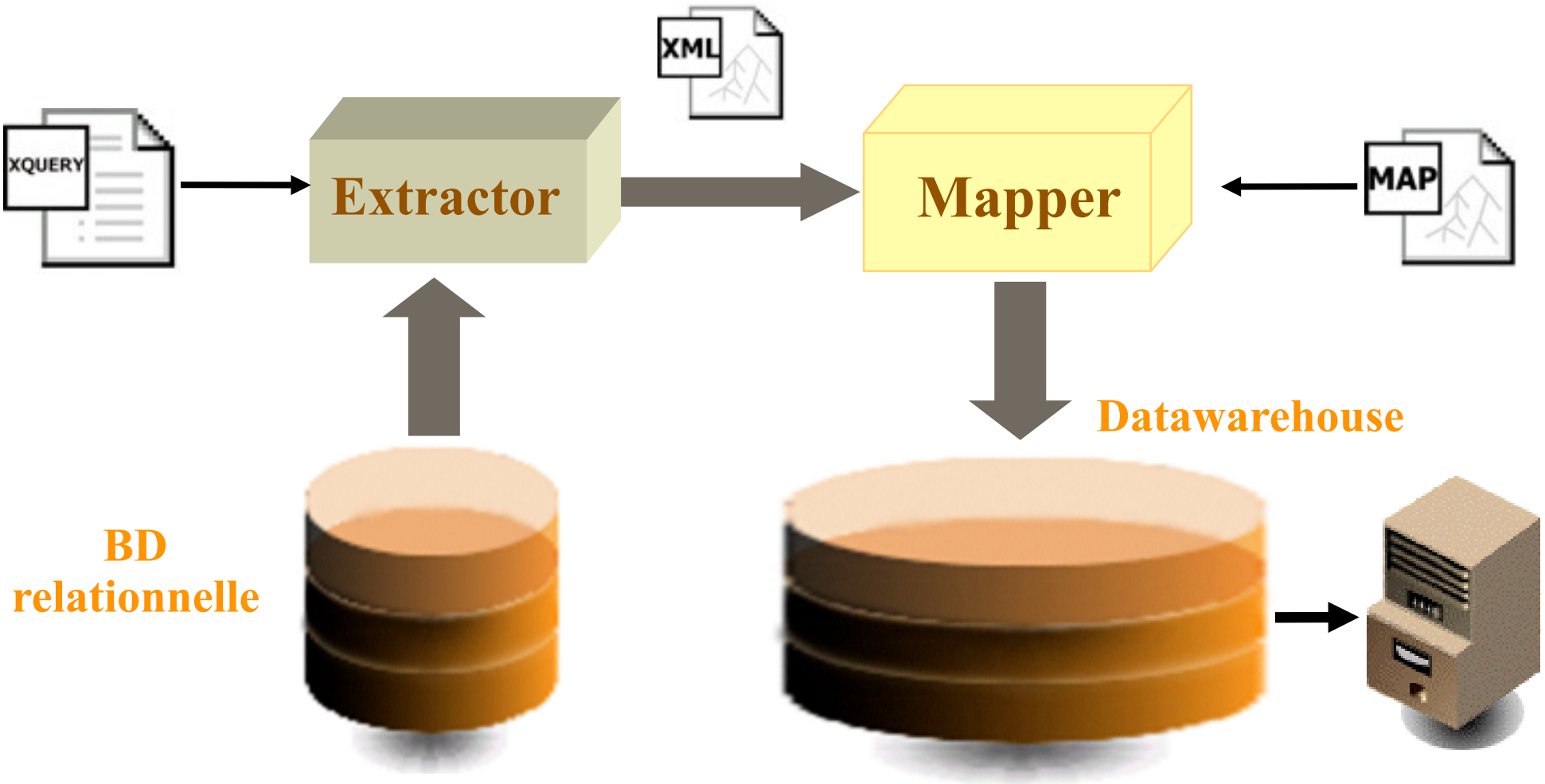
- Un standard d'échange
  - Lisible : texte balisé avec marquage
  - Clair : séparation du fond et de la forme
  - Extensible : supporte les évolutions applicatives
  - Sécurisé : pare-feu, encryption, signature
- Développé par le W3C
  - Pour le Web (Internet, Intranet)
  - S'étend à l'entreprise et ses partenaires
- Supporté par les grands constructeurs
  - IBM, Microsoft .net, SUN, BEA, etc.
  - Des outils génériques et ouverts

Publication multi-supports



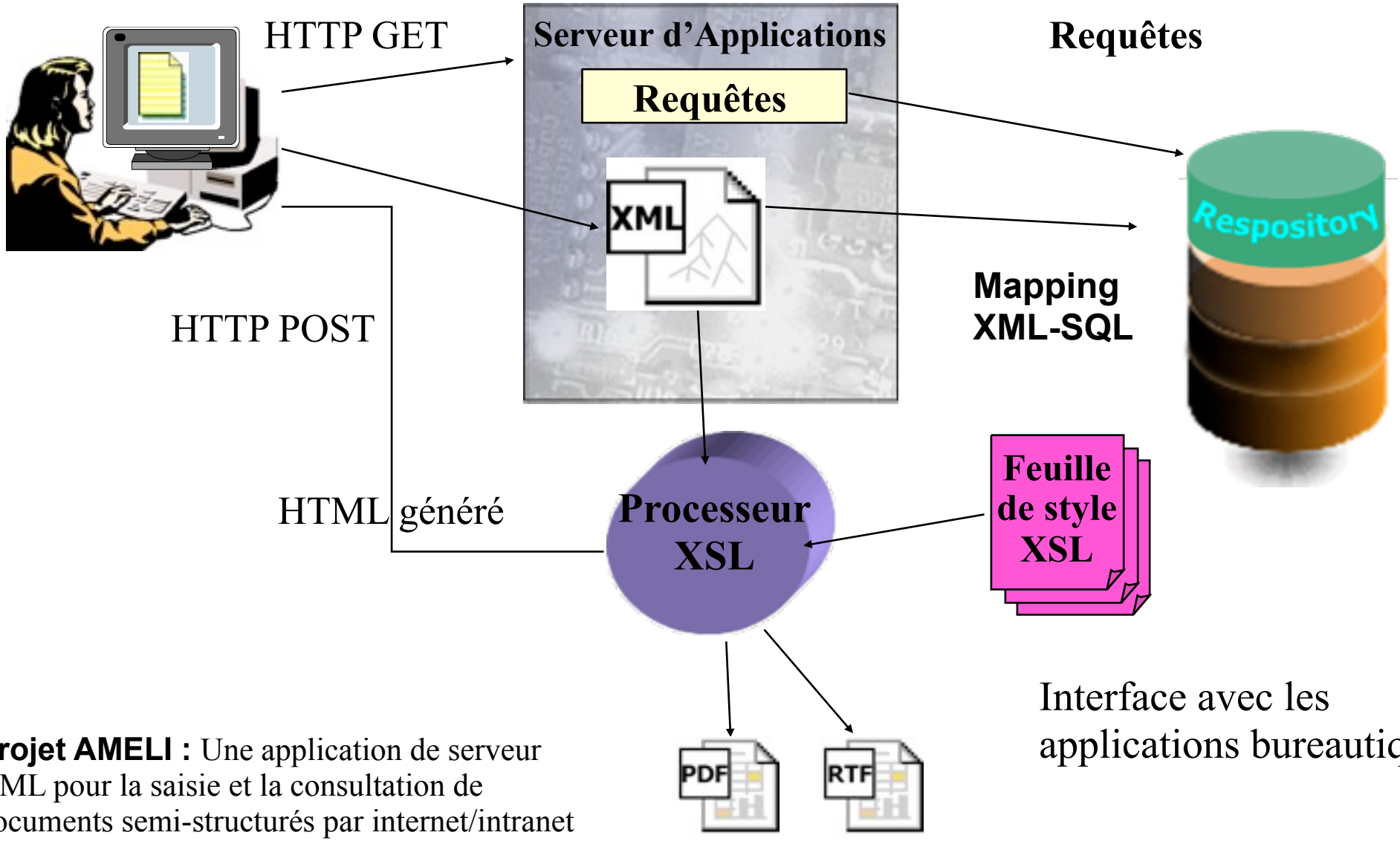
Projet GIE Cartes Bancaires:
publication des spécifications de protocoles pour réseau Carte Bancaires vers les fournisseurs et prestataires du GIE

Échange de données: ETL



Projet MEN : Echange de données avec les académies (remontée et descente)

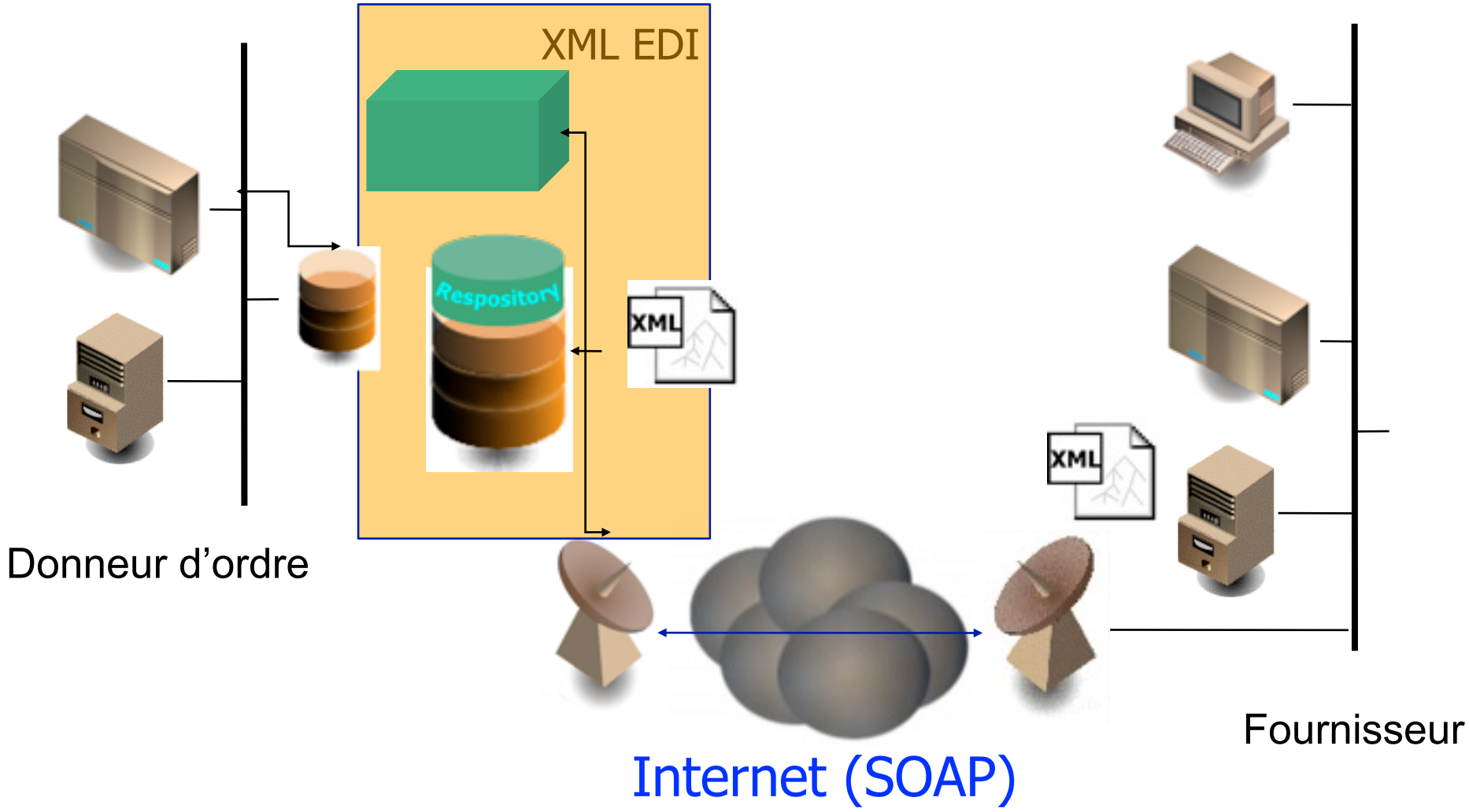
Gestion documents semi-structurés



Projet AMELI : Une application de serveur XML pour la saisie et la consultation de documents semi-structurés par internet/intranet

Interface avec les applications bureautiques

Échange B2B XML EDI



Projet EDIXML : Expérimentation de la chaîne pour un grand donneur d'ordres

Forces et faiblesses de XML

- Une technologie structurante
- Clarifie tous les échanges
- Des standards internes et externes
- Transversale à l'entreprise
  - Échanges de données
  - Bureautique
  - GED
  - Sites Web
  - EDI
  - Bases de données
  - Intégration e-business
  - …

- Une syntaxe bavarde
- Un méta-langage, mais de nombreux langages
- Coûteux en CPU
  - Parsing
- Coûteux en mémoire
  - Instanciation
- Format compressé à l'étude

Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

Syntaxe de base

Ou comment réaliser des documents bien formés

Concepts du modèle

- Balise (ou tag ou label)
  - Marque de début et fin permettant de repérer un élément textuel
  - Forme: <balise> de début, </balise> de fin
- Elément de données
  - Texte encadré par une balise de début et une de fin
  - Les éléments de données peuvent être imbriquées (donc un élément peut contenir des balises)
    - <producteur>
    - <adresse>
    - <rue>A. Briand</rue>
    - <ville>Dijon</ville>
    - </adresse>
    - </producteur>
- Attribut
  - Doublet nom="valeur" qualifiant une balise
    - <producteur no="160017" region="Bourgogne">

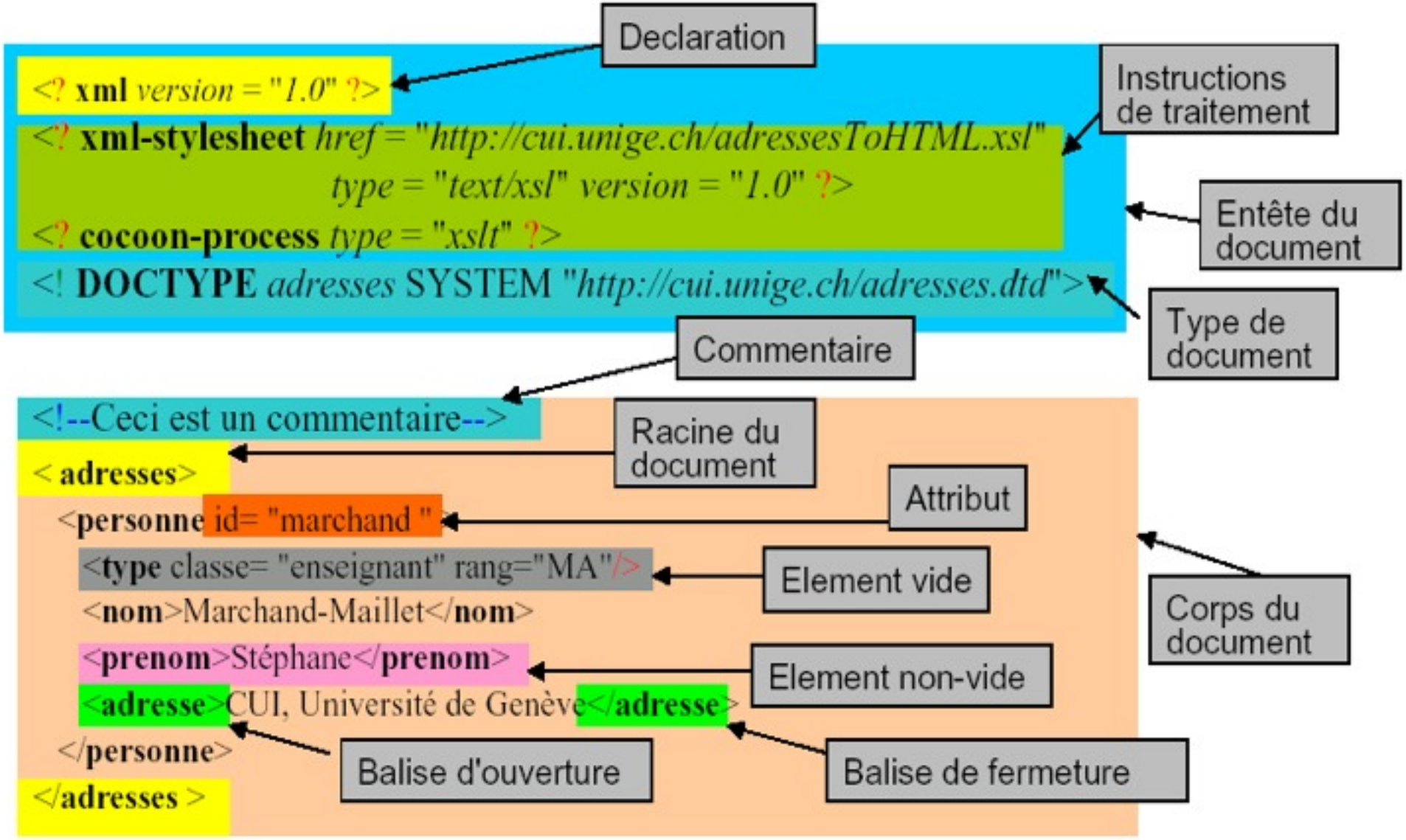
Quelques conventions

- Le nom d'une balise est formé des lettres [a...bA...B], de chiffres ou des symbols “- _ .”
- <explication xml:lang="fr">
- <lavie></lavie> est équivalent à <lavie/>

Quelques règles de syntaxe

- Un document XML ne doit avoir qu'un et un seul élément appelé « élément-racine », qui doit contenir tous les autres éléments.
- Pas de chevauchement: <a> n'est pas autorisé.
- L'ordre des éléments a de l'importance.
- L'ordre des attributs n'en a pas.
- On peut écrire des commentaires à l'extérieur des balises en les encadrant de <!-- *comment* -->

Exemple complet



Entités

- On ne peut pas utiliser certains caractères dans le texte des éléments. Ex: < et &
- Si on en a besoin, on réalise alors un appel d'entités: &nom_de_l_entité;
- Quelques entités prédéfinies:

< < & & > > " " ' '

- On peut aussi utiliser le segment <![CDATA[j'&cris_ce_que <je_veux>]]>

Information de traitement

- XML ne permet pas d'appeler directement des programmes (impossible de générer une partie du texte par une application externe).
- Par contre on peut utiliser des informations de traitement permettant d'indiquer aux programmes comment trouver un certain résultat:
  <?nom-programme contenu ?>
- On peut les placer partout sauf dans les balises.
- La syntaxe doit être du XML valable.

La déclaration XML

- Ressemble à une information de traitement : <?xml ... ?>
- Contient au plus trois attributs :
  - version="..." (presque toujours “1.0”),
  - encoding="..." (pour nous souvent "ISO-8859-1"),
  - standalone="..." (valant “yes” ou “no” suivant si on veut utiliser une DTD externe ou non).
- La déclaration XML doit obligatoirement être au tout début du document (même un espace n'est pas permis !) ou bien être absente.

Plan du cours

- Avant de commencer
- Un peu d'histoire
- Quelques définitions
- Pourquoi le XML
- Syntaxe de base
- Définition de Type de Documents

Définition de Type de Document

Un document XML peut être associé à :

- une DTD ou un schéma XML pour décrire les balises autorisées (et leur structure)
- une feuille de style pour présenter les données

DTD et schéma XML permettent de définir son propre langage basé sur XML :

- Vocabulaire (balises)
- Grammaire (règles d'imbrication)

➔ Dialecte ou Jargon

Les DTD

- Les DTD sont des fichiers spécifiant le vocabulaire et la structure d'un type de documents XML.
- But : rajouter des contraintes à la forme des éléments et attributs (nom, ordre, contenu).
- Un document XML *bien formé* suivant une DTD donnée est appelé document valable.
- Supporté par la plupart des logiciels (java, firefox, ...)
- Ce n'est pas du XML.
- On définit rarement une DTD à partir de rien.

Utilisation d’une DTD externe

<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>

<!DOCTYPE baliseracine SYSTEM “http://www.toto.com/madtd.dtd">

- standalone=”no” : pour dire au logiciel d'aller chercher une DTD externe.
- baliseracine : nom de la balise racine définie dans la DTD.
- http://etc. : emplacement (global ou local) où trouver la DTD.

Utilisation d'une DTD interne

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE baliseracine [
<!ELEMENT baliseracine (PERSONNE*)>
<!ELEMENT PERSONNE (#PCDATA)>
<!ATTLIST PERSONNE PNUM ID #REQUIRED>
]>
<baliseracine>
</baliseracine>
```

- standalone="yes" : pour dire au logiciel que la DTD est définie dans le document.
- DOCTYPE : mot-clé correspondant au début de la définition de la DTD.

DTD externe, interne

- Externe : en spécifiant l'adresse de la DTD.
- Interne : directement en entête du fichier XML.
- Combinaison des deux :

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
  <!DOCTYPE problemset SYSTEM "../dtd/extdtd.dtd"[
  <!ELEMENT question #PCDATA>

    ...

  ]>
```

La DTD interne est lue avant la DTD externe, il ne doit pas y avoir de conflit.

DTD : syntaxe des éléments

- <!ELEMENT balise (définition) > permet de définir ce qui peut être contenu dans un élément de nom “balise”.
- Le paramètre *définition* représente soit un type de donnée prédéfini, soit un élément de données composé, constitué lui même d'éléments
- Types prédéfinis :
    - ANY : L'élément peut contenir tout type de donnée
    - EMPTY : L'élément ne contient pas de données spécifiques
    - #PCDATA : L'élément doit contenir une chaîne de caractère
- Utilisation :
    - ANY et EMPTY : sans parenthèses
    - (#PCDATA) : avec parenthèses

DTD : syntaxe des éléments

Déclaration d'éléments composés :

Exemple : <!ELEMENT montant (devise?,valeur) >

Opérateur	Signification	Exemple
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
|	L'élément A **ou** B peuvent être présents (pas les deux)	A|B
,	L'élément A doit être présent et suivi de l'élément B	A,B
()	Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs	(A,B)+

Attention à l'ordre !

<!ELEMENT mont (devise, somme) > ≠ <!ELEMENT mont (somme, devise) >

Exemple de définition d'éléments

- Exemple de DTD :

```
<!ELEMENT personne (nom, prenom+, tel?, adresse) >
<!ELEMENT nom        (#PCDATA) >
<!ELEMENT prenom     (#PCDATA) >
<!ELEMENT tel        (#PCDATA) >
<!ELEMENT email      (#PCDATA) >
<!ELEMENT adresse       ANY       >
```

- Exemple de document :

```
<personne>
   <nom>Hugo</nom>
   <prenom>Victor</prenom>
   <prenom>Charles</prenom>
   <tel>01120243</tel>
   <adresse><rue></rue><ville>Paris</ville></adresse>
 </personne>
```

DTD : syntaxe des attributs

- Par défaut, aucun attribut aux éléments.
- <!ATTLIST nom_element nom_attribut type mode>
- Le mode précise si l'attribut est obligatoire (#REQUIRED) ou optionnel (#IMPLIED), ou fixé (#FIXED).

<!ATTLIST homme age CDATA #IMPLIED> signifie qu'un élément “homme” peut avoir un attribut “age” textuel.

<!ATTLIST homme age CDATA #FIXED "40 ans">

<!ATTLIST piece position (face|pile) #REQUIRED>

- Pour associer plusieurs attributs à un élément :
  - spécifier plusieurs lignes “<!ATTLIST ...>”
  - ou en mettre plusieurs à la fois (ordre sans importance).

Types des Attributs

Le type de valeur peut être :

- `CDATA` : texte,
- `ID` : nom XML (et non pas token de nom), identifiant l'élément dans le document. La valeur d'un attribut de type ID ne peut pas être utilisée deux fois dans le même document et il y a au plus un attribut de type ID par élément
- `IDREF` ou `IDREFS` : identificateur ou suite d'identificateurs (séparés par une suite de séparateurs : espace, CR, LF, tabulation),
- `NMTOKEN` ou `NMTOKENS` : token de nom ou suite de tokens de nom, $(nom_1 \ \ldots \ nom_n)$
- *`Enumeration:`* $(nom_1 \mid \ldots \mid nom_n)$

Exemples d'attributs

```
<! ATTLIST personne
                num ID                              #REQUIRED
                age CDATA                           #IMPLIED
                genre (Masculin | Feminin )    #REQUIRED
    >

<!ELEMENT auteur (#PCDATA) >

<!ELEMENT editeur (#PCDATA) >

<!ATTLIST auteur
                genre (Masculin | Feminin )    #REQUIRED
                ville CDATA                         #IMPLIED
    >

<!ATTLIST editeur
                 ville CDATA #FIXED "Paris">
```

Exemple de ID et IDREF

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
<!ELEMENT DOCUMENT(PERSONNE*)>
<!ELEMENT PERSONNE (#PCDATA)>
<!ATTLIST PERSONNE PNUM ID #REQUIRED>
<!ATTLIST PERSONNE MERE IDREF #IMPLIED>
<!ATTLIST PERSONNE PERE  IDREF #IMPLIED>
]>
<DOCUMENT>
<PERSONNE PNUM = "P1">Marie</PERSONNE>
<PERSONNE PNUM = "P2">Jean</PERSONNE>
<PERSONNE PNUM = "P3" MERE="P1" PERE="P2">Pierre</PERSONNE>
<PERSONNE PNUM = "P4" MERE="P1" PERE="P2">Julie</PERSONNE>
</DOCUMENT>
```

DTD : notion d'entités

- <!ENTITY % mon_entite "#PCDATA|age*">

Cela signifie que partout "%mon_entite" sera remplacé par "#PCDATA|age*"

- Ces entités paramètres doivent être définies dans une DTD externe (elles ne sont pas prises en considération si elles sont définies en interne).

Quelques règles d'écriture

- Modularité
  - définir dans des entités séparées les parties réutilisables
- Précédence
  - Regrouper les déclarations d'entités en tête
- Abstraction
  - Utiliser des entités pour les modèles de contenus
- Spécificité
  - Éviter les DTD trop générales
- Simplicité
  - Découper les DTD trop complexes

Insuffisance des DTD

- Pas de types de données
  - difficile à interpréter par le récepteur
  - difficile à traduire en schéma objets
- Pas en XML
  - langage spécifique
- Propositions de compléments
  - XML-data de Microsoft (BizTalk)
  - XML-schema du W3C