

Intégration et publication avec XML



Publication

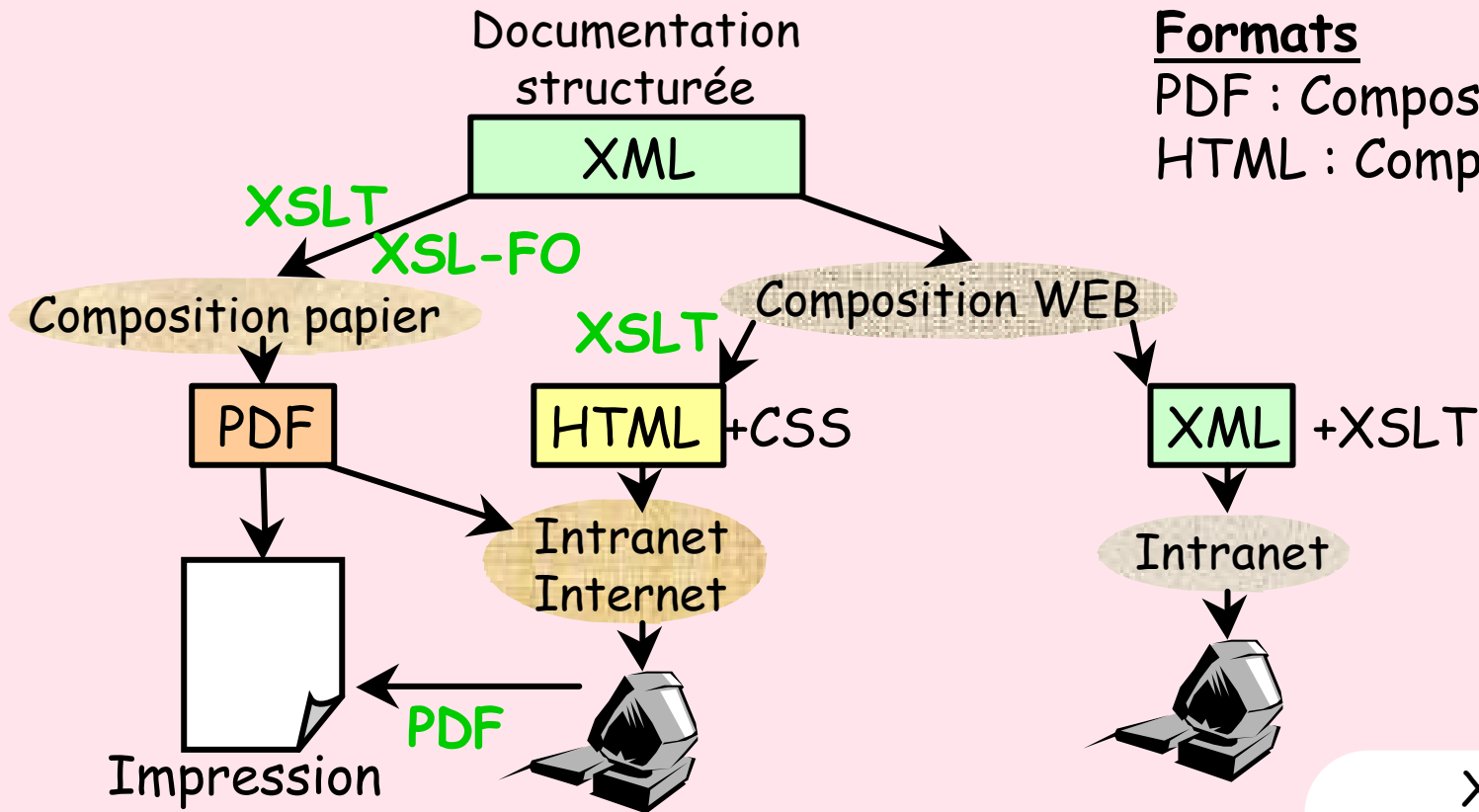
- Publication cross-media
 - Modes de publications avec XSLT
 - Transformations côté serveur, late-binding, profiling
 - Architecture
-

Integration

- Agrégation de données
 - Distribution de données
 - Granularité
 - Applications documentaires, XHTML, DocBook
 - RDF, dublin-core, syndication, RSS
-

Applications XML

- Typologie
- Cocoon : un framework de publication
- Integration avec les Services Web
- L'architecture MVC
- Configurations en XML



Formats

PDF : Composition papier
HTML : Composition Web

Source : unique, neutre (XML)

Cible : multiple, hétérogène

Composition papier : format PDF

Publication électronique : à base de XML

Intranet : XML (natif) + XSLT

Internet : HTML (XML + XSLT + scripts) + CSS

Bureautique : prévue pour faire du papier. Inadaptée?

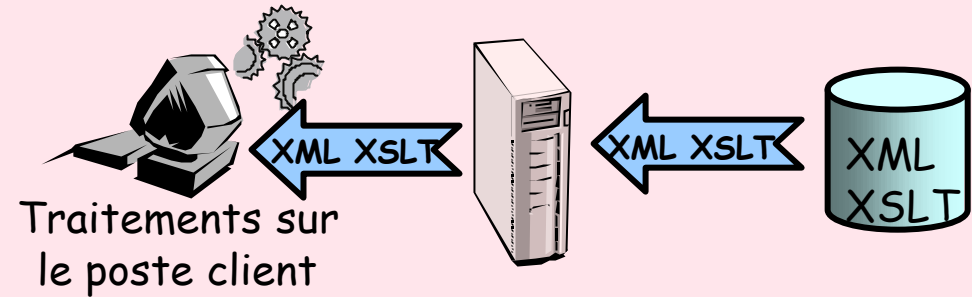
XML :
write once,
publish everywhere

XML est orienté **structure**.
Pour le publier on utilise un
procédé de transformation
approprié au média de
diffusion : **XSLT**
(eXtensible **S**tylesheet
Language - Tranformations)

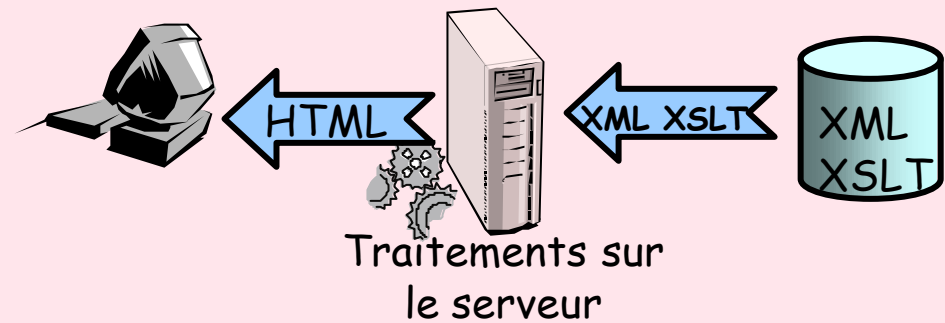
XSLT est un langage de
programmation qui utilise
une grammaire XML.

3 méthodes pour
disposer d'un format
consultable

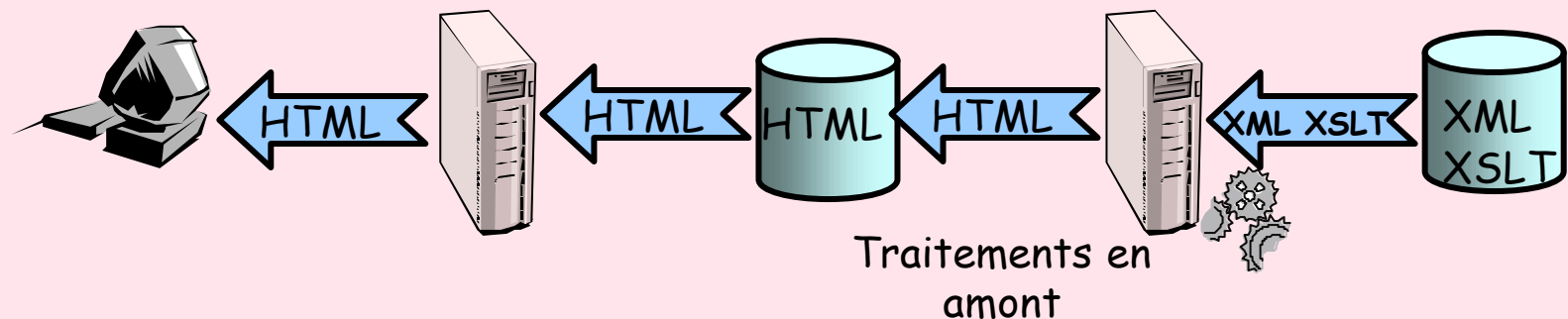
Transformation sur le poste client



Transformation à la volée sur le serveur



Transformation et stockage sur le serveur



Transformation statique

Avantages :

- Utilisation d'un serveur Web classique
- Publication à emporter (sur cédérom)
- Snapshot du fonds documentaire
- Production des publications en Batch
- Robustesse

Inconvénients :

- Gestion des publications
- Stockage des publications

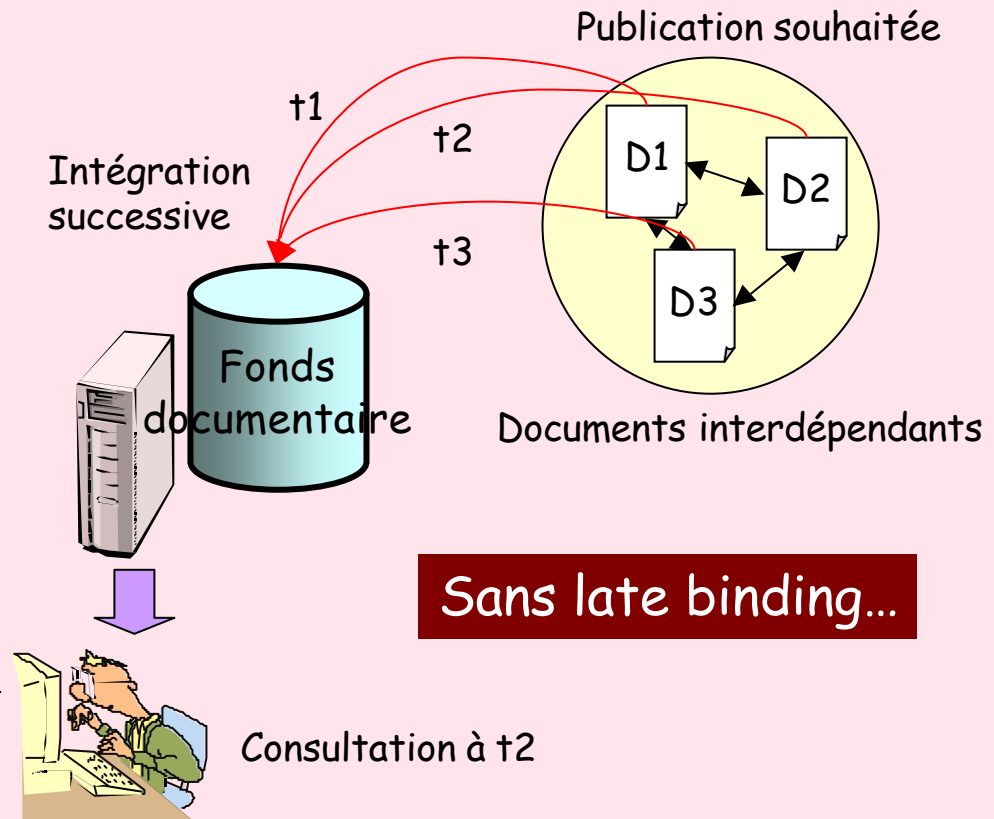
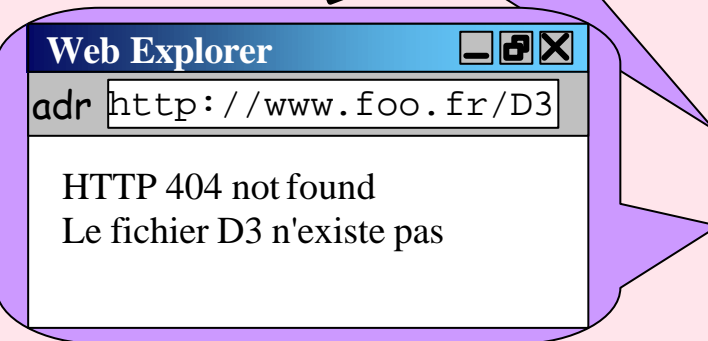
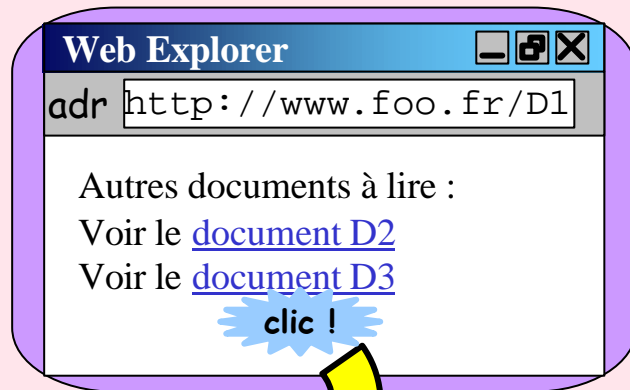
Transformation à la volée

Avantages :

- Late binding
- Profiling possible
- Interactivité plus grande
- Pas de gestion des fichiers produits
- Qui peut le plus peut le moins

Inconvénients :

- Publication non portable
- Attention aux temps de réponse
- Requier un serveur robuste



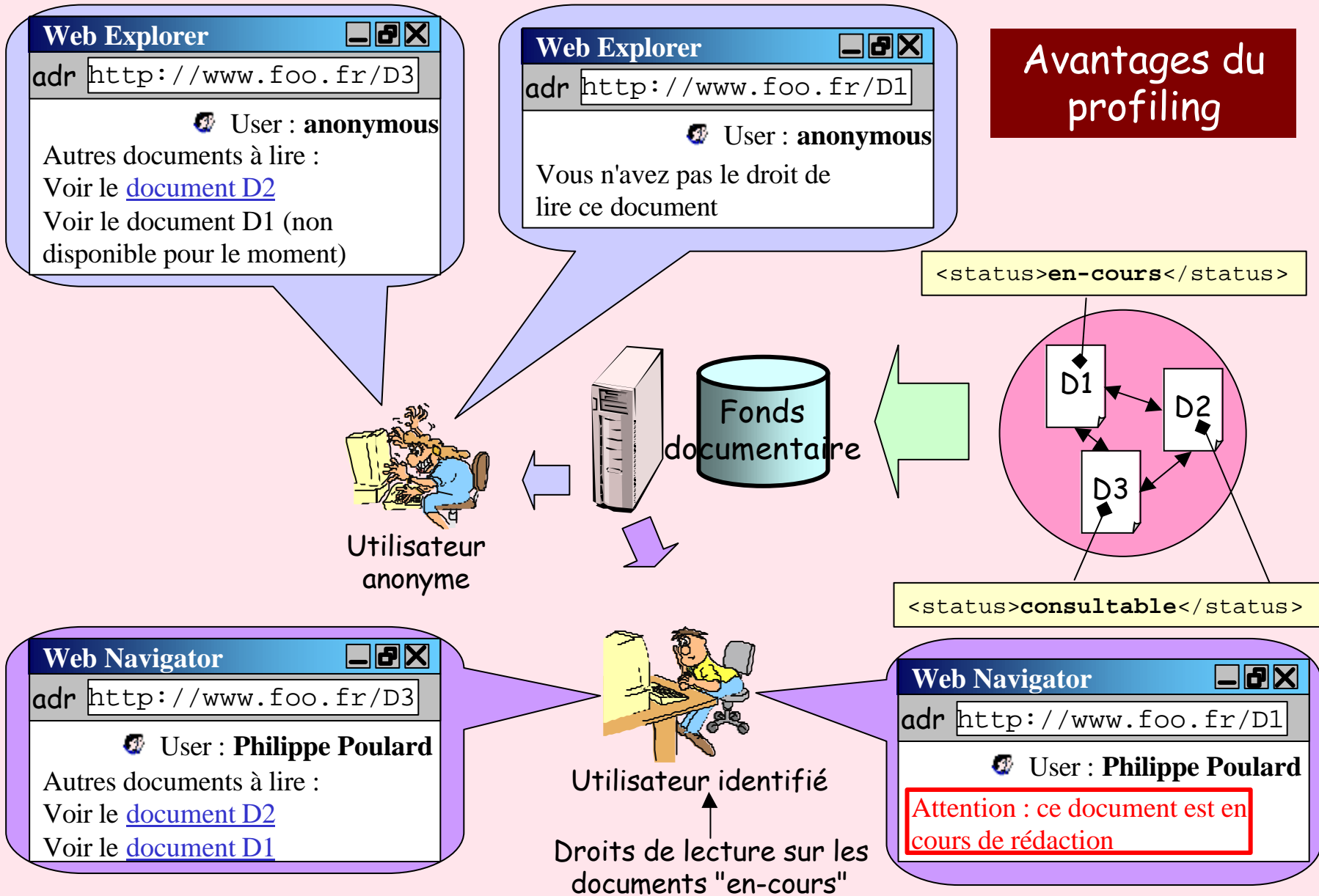
Si les documents sont intégrés les uns après les autres, il y aura, selon la stratégie de publication :

- des liens non résolus
- ou des liens cassés

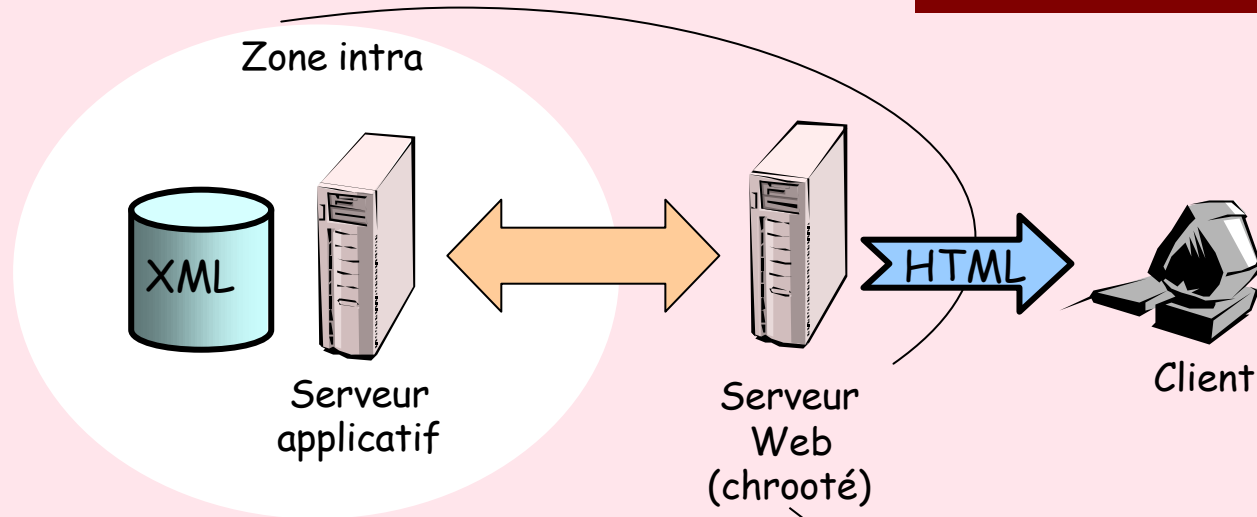
ou

Nécessité de générer la publication :

- à chaque fois qu'un document est intégré
- ou à la fin quand tous les documents ont intégrés le fonds documentaire



Architecture multi-tiers



Le serveur applicatif délivre les documents que lui réclame le serveur Web
Il indique au serveur Web de vider le cache d'un document lorsque celui-ci a été modifié

Le serveur Web vérifie s'il dispose du document dans son cache.
S'il ne l'a pas, il le demande au serveur d'application, puis le met dans son cache

Au cœur du serveur Web



Serveur
Web
(chrooté)

Quand un document est demandé par un client :

- on le cherche dans les différents niveaux de cache
- s'il n'est pas dans un cache, on le demande au serveur d'application
- il est chargé dans le DOM
- le granule demandé est transformé par XSLT
- on délivre au client le résultat attendu
- pendant ce temps, on anticipe plusieurs compositions (page suivante, version PDF...)

Un document subit plusieurs étapes de transformation :

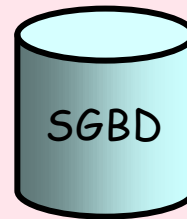
- Les PI qui sont traitées
- Les liens logiques sont résolus en fonction du profil de l'utilisateur
- Le résultat est transformé par XSLT

Multi-level caching :

- un document qui vient d'être sollicité est conservé en mémoire en objet session DOM (car les liens dépendent du profil de l'utilisateur)
- la version d'origine est conservée en mémoire en référence faible DOM (si la mémoire manque, le système libère la place occupée)
- le document provenant du serveur d'application est stocké sur disque

Sources :
• XML
• non XML

Données stockées



Flux



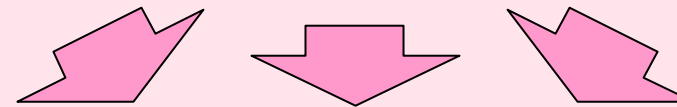
Application



PGI

Intérêt

- Séparation des points de vues (les préoccupations des web-designers ne sont pas les mêmes que celles des programmeurs)
- Découpler la logique métier de la logique de présentation
- Intéropérabilité
- Universalité

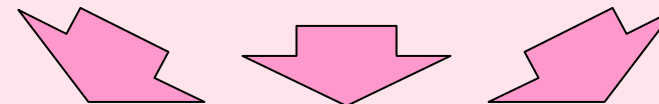


XML

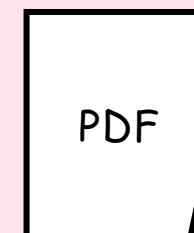
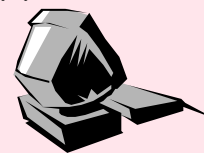
```

<?xml version="1.0"
encoding="ISO-8859-1" ?>
<!DOCTYPE rapport
SYSTEM "rapport.dtd" [
<ENTITY ventes
SYSTEM "ventes.xml">
]>
<rapport>
  &ventes;
  <analyse>
    <p>Cette année, la croissance
est au rendez-vous,
.../...
  </p>
  </analyse>
</rapport>

```



Application

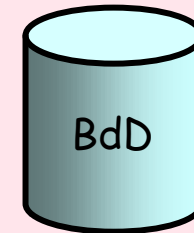


Exemple : construction d'un rapport

Edition à partir d'un éditeur XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE rapport SYSTEM "rapport.dtd" [
  <!ENTITY ventes SYSTEM "ventes.xml">
]>
<rapport>
  &ventes;
  <analyse>
    <p>Cette année, la <b>croissance</b>
      est au rendez-vous,
      .../...
    </p>
  </analyse>
</rapport>
```

Système d'information

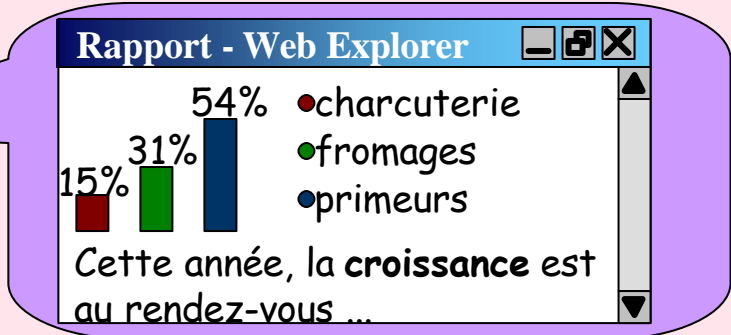
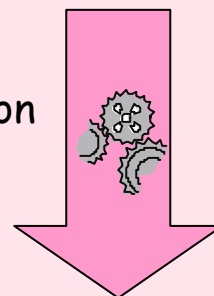


Génération automatique du fichier

ventes.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ventes>
  <charcuterie>23</charcuterie>
  <fromages>45</fromages>
  <primeurs>78</primeurs>
</ventes>
```

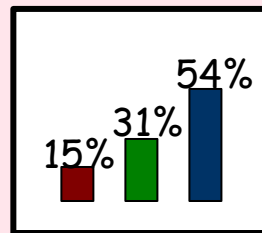
Publication



```
<?xml version="1.0"
encoding="ISO-8859-1" ?>
<!DOCTYPE rapport
SYSTEM "rapport.dtd" [
<ENTITY ventes
SYSTEM "ventes.xml">
]>
<rapport>
&ventes;
<analyse>
<p>Cette année, la croissance
est au rendez-vous,
.../...
</p>
<analyse>
</rapport>
```



```
<svg>
</svg>
```



```
<html xmlns="http://www.w3.org/TR/xhtml1/strict">
<head>
<title>Rapport des ventes 2002</title>
</head>
<body>
<h1>Rapport</h1>
<p>
<table align="right">
<tr><td>charcuterie</td></tr>
<tr><td>fromages</td></tr>
<tr><td>primeurs</td></tr>
</table>
</p>
<p>Cette année, la <b>croissance</b> est au rendez-vous ...</p>
</body>
</html>
```



Rapport - Web Explorer

Rapport des ventes 2002

- charcuterie
- fromages
- primeurs

15% 31% 54%

Cette année, la **croissance** est au rendez-vous ...

Quelle est la meilleure stratégie ?

- 1 document source (XML) → 1 document publié (HTML)
- 1 document source (XML) → n documents publiés (HTML)
- n documents sources (XML) → 1 document publié (HTML)
- n documents sources (XML) → m documents publiés (HTML)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
.../...
<content>
  <section><title>Objectifs</title>
  <p>Ce document décrit comment
    monter et démonter un blah.
  </p>
</section>
  <section><title>Prérequis</title>
  <p>Le lecteur doit avoir une
    connaissance des blah approfondie.
  </p>
  <p>En particulier, il doit maîtriser
    les blah bleus et les blah
    de compétition.
  </p>
</section>
  <section><title>Manipulation</title>
  <info>Penser à vérifier les blah
    avant toute manipulation.
  </info>
  <step>Enlever l'eau sale du blah.</step>
  <step>Dévisser le capot inférieur.</step>
  <step>Cliquer sur OK.</step>
</section>
.../...
```

Blah - Web...

Les blah

Sommaire :

- 1 [Objectifs](#)
- 2 [Prérequis](#)
- 3 [Manipulation](#)

Blah - Web Explorer

1 - Objectifs [\[Sommaire\]](#) [\[Page 2\]](#)

Ce document décrit comment monter et démonter un blah.

Blah - Web Explorer

2 - Prérequis [\[Page 1\]](#) [\[Sommaire\]](#) [\[Page 3\]](#)

Le lecteur doit avoir une connaissance des blah approfondie. En particulier, il doit maîtriser les blah bleus et les blah de compétition.

Blah - Web Explorer

3- Manipulation [\[Page 2\]](#) [\[Sommaire\]](#)

Penser à vérifier les blah avant toute manipulation.

- Enlever l'eau sale du blah.
- Dévisser le capot inférieur.
- Cliquer sur OK.

XHTML

Les parties narratives des documents XML pourraient être basées sur HTML

- permet de réutiliser ce qui existe déjà

XHTML est :

- une reformulation de HTML en XML
- une conception **modulaire** de HTML

Permet de "piocher" dans XHTML les besoins strictement nécessaires

Proscrire l'utilisation seule de XHTML (on perd tout les avantages de XML)

Comme HTML, XHTML ne devrait pas être utilisé en tant que format source

XHTML est un format de présentation pour le Web

Dans les documents sources, à réserver pour les parties narratives d'un document

Document source XML

```
<?xml version="1.0"
  encoding="ISO-8859-1" ?>
<rapport>
  <ventes>
    <charcuterie>23</charcuterie>
    <fromages>45</fromages>
    <primeurs>78</primeurs>
  </ventes>
  <analyse>
    <p>Cette année, la <b>croissance</b>
    est au rendez-vous,
    .../...
  </p>
  </analyse>
</rapport>
```

Partie narrative
en XHTML

DocBook

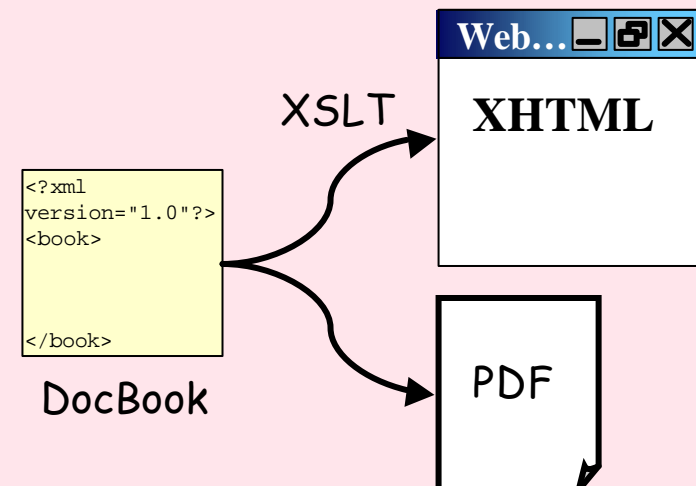
Autre standard "concurrent" : DocBook

- beaucoup plus formel
- spec de 700 pages
- de très nombreux éléments
- orienté présentation documentaire
- un peu lourd (appropriation difficile)

<http://www.docbook.org>

Peut servir pour des documents sources faiblement structurés

Usage : comme avec XHTML, on peut construire ses propres DTD en s'inspirant de ce qui est fait dans DocBook



Fermer les balises `Foo` → `Foo`

Imbriquer correctement les balises `<p><i>Foo</p></i>` → `<p><i>Foo</i></p>`

Casse des balises et attributs en minuscule `<CODE>Foo</code>` → `<code>Foo</code>`

Guillemets autour des valeurs des attributs `<p align=center>` → `<p align="center">`

Assigner une valeur à tous les attributs

`<input type="checkbox" checked>` → `<input type="checkbox" checked="checked">`

Echapper & et < par les entités `&` et `<`;

`Foo` → `Foo`

Un seul élément racine html

Fermer les balises vides `<hr>` → `<hr/>`

Déclarer l'encodage utilisé `<?xml encoding="ISO-8859-1"?>`

Ajouter une déclaration DOCTYPE avec un identificateur PUBLIC

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "xhtml1-strict.dtd">`

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "xhtml1-transitional.dtd">`

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "xhtml1-frameset.dtd">`

Déclarer l'espace de noms xhtml sur l'élément racine

`<html xmlns="http://www.w3.org/1999/xhtml">`

XHTML : compatibilité avec HTML

Trucs pour les anciens navigateurs...

Contraction de la
fermeture des balises`
` Non`
` Oui

...À éviter parfois

`<input ... />` Non`<input ...></input>` Oui

Eviter d'inclure des instructions de traitement : elles risquent d'être affichées

Les interpréteurs JavaScript ou VBScript risquent de ne pas reconnaître `&` et `<`, qu'on trouve souvent... utiliser des scripts externes.

Préférer `"` à `'`

Strict

Forme recommandée par le W3C.

Ne contient pas les éléments et attributs dépréciés.

Interdit l'usage des attributs de présentation, qui doivent être pris en charge par CSS

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "DTD/xhtml11-strict.dtd">
```

Transitional

Contient les éléments et attributs dépréciés.

Ne contient pas les éléments relatifs aux cadres (frameset, iframe...)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "DTD/xhtml11-transitional.dtd">
```

Frameset

Identique à Transitional, avec en plus les éléments relatifs aux cadres

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "DTD/xhtml11-frameset.dtd">
```

XHTML 1.1 fragmente les DTD en modules appelables via des entités paramètres.

```
%html-struct.module;
```

html, head, title **et** body

```
%html-text.module;
```

abbr, acronym, address, blockquote, br, cite, code, dfn, div, em,
h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var

```
%html-hypertext.module;
```

a

etc...

Utilisation

Dans vos DTD

```
<!ENTITY % xhtml-hypertext.module "INCLUDE" >  
<![%xhtml-hypertext.module;  
    <!ENTITY % xhtml-hypertext.mod PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Hypertext//EN"  
        "xhtml-hypertext-1.mod">  
    %xhtml-hypertext.mod;  
]]>
```

Web Sémantique

Resource Description Framework

Permet de décrire des dépendances entre des ressources

C'est un framework : peut être décliné de différentes façons selon les domaines d'application

Méta-données : ce ne sont pas les données qui relient les différentes ressources, mais les informations qui décrivent ces données

Utilité :

- Indexation
- Statistique
- Gestion
- Analyse
- Recherche

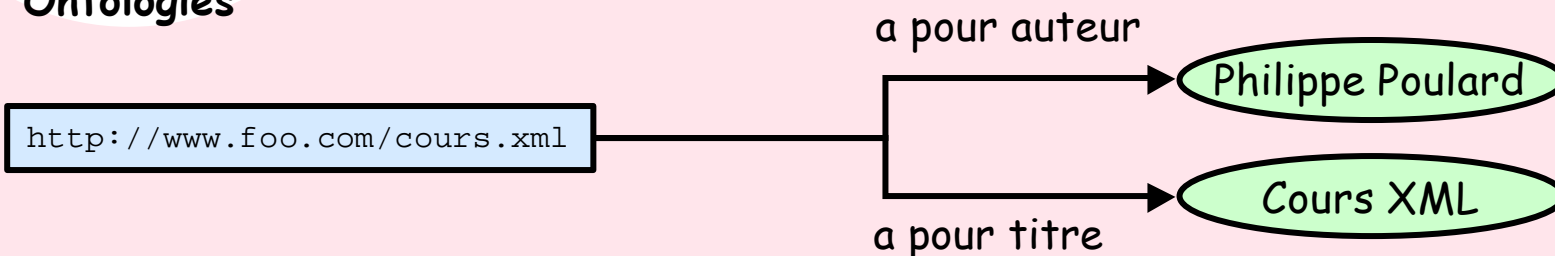
Un vocabulaire RDF définit un ensemble d'éléments et le contenu pour les méta-données de chaque domaine.



Ontologies

Dublin-core : vocabulaire RDF qui définit les méta-données sur le contenu des documents du Web

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/DC/"
  <rdf:Description about="http://www.foo.com/cours.xml">
    <dc:CREATOR>Philippe Poulard</dc:CREATOR>
    <dc:TITLE>Cours XML</dc:TITLE>
  </rdf:Description>
</rdf:RDF>
```



TITLE	Nom de la ressource
CREATOR	Personne ou organisation créatrice de la ressource
SUBJECT	Sujet de la ressource
DESCRIPTION	Description de la ressource (résumé)
PUBLISHER	L'éditeur de la ressource
CONTRIBUTOR	Personne ayant collaboré à la création de la ressource
DATE	Date de publication de la ressource
TYPE	Type de la ressource (page Web, poème, photographie...)
FORMAT	Format de la ressource (PDF, HTML, JPEG...)
IDENTIFIER	Identifiant unique de la ressource (ISBN, URI...)
SOURCE	Information indiquant que la ressource est dérivée d'une autre source
LANGUAGE	Code langue ISO 639 dans lequel la ressource est rédigée
RIGHTS	Note de droits de copie et de propriété intellectuelle

RSS : Rich Site Summary

RSS est une application RDF

Permet d'enrichir ses propres listes de publications de celles des autres, et de faire partager les siennes aux autres : d'où l'utilité d'utiliser un vocabulaire commun

```
<rdf:RDF
  xmlns="http://purl.org/rss/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:synd="http://purl.org/rss/1.0/modules/syndication">
<channel rdf:about="http://www.foo.com/cours-rss.rdf">
  <title>Cours XML</title>
  <description>De nombreux cours sur les technologies XML</description>
  <dc:creator>Philippe Poulard</dc:creator>
  <dc:publisher>Philippe Poulard</dc:publisher>
  <synd:updatePeriod>monthly<synd:updatePeriod>
  <synd:updateFrequency>1<synd:updateFrequency>
  <items>
    <rdf:Seq>
      <rdf:li resource="http://www.foo.com/cours-xml/core.xml"/>
      <rdf:li resource="http://www.foo.com/cours-xml/dom.xml"/>
      <rdf:li resource="http://www.foo.com/cours-xml/xpath.xml"/>
      <rdf:li resource="http://www.foo.com/cours-xml/xslt.xml"/>
    </rdf:Seq>
  </items>
</channel>
```

.../...

```
.../...
```

```
<item rdf:about="http://www.foo.com/cours-xml/core.xml">
  <title>Cours XML</title>
  <source url="http://www.foo.com/cours-xml/core.xml"/>
  <dc:date>28/11/2002</dc:date>
  <description>Cours sur XML : syntaxe, DTD, espaces de nommage</description>
</item>
<item rdf:about="http://www.foo.com/cours-xml/dom.xml">
  <title>Cours DOM</title>
  <source url="http://www.foo.com/cours-xml/dom.xml"/>
  <dc:date>28/11/2002</dc:date>
  <description>Cours de programmation avec l'API DOM pour XML</description>
</item>
<item rdf:about="http://www.foo.com/cours-xml/xpath.xml">
  <title>Cours XPath</title>
  <source url="http://www.foo.com/cours-xml/xpath.xml"/>
  <dc:date>28/11/2002</dc:date>
  <description>Cours sur le langage d'adressage XPath</description>
</item>
<item rdf:about="http://www.foo.com/cours-xml/xslt.xml">
  <title>Cours de programmation avec XSLT</title>
  <source url="http://www.foo.com/cours-xml/xslt.xml"/>
  <dc:date>28/11/2002</dc:date>
  <description>Cours sur le langage de programmation XSLT</description>
</item>
</rdf:RDF>
```

Typologie

Applications de présentation :

- XHTML
- WML
- XSLFO
- SVG
- SMIL

Applications d'échange :

- SOAP
- UDDI

Applications sectorielles :

- MathML
- CML
- DocBook

Applications de description :

- RDF
- RSS
- WSDL

Exemple d'architecture évoluée :

- Web services

Exemples d'application à haut niveau de service :

- Portail documentaire

Intégration de nombreuses sources :

- Documents XML
- SGBD
- LDAP



Négociation de contenu avec le client : fourni des sorties WML, XHTML, XML, PDF...

Utilise une technologie alternative aux technologies côté serveur (ASP, PHP, JSP) : XSP(XML Server Page), « mélange » de Java encapsulé dans XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Stylesheet for web browsers -->
<?xml-stylesheet href="rapport-html.xsl" type="text/xsl"?>
<!-- Stylesheet for Java clients -->
<?xml-stylesheet href="rapport-xml.xsl" type="text/xsl" media="java"?>
<!-- Processing instructions for Cocoon -->
<?cocoon-process type="xsp"?>
<?cocoon-process type="xslt"?>
<!-- XSP -->
<xsp:page xmlns:xsp="http://www.apache.org/1999/XSP/Core"
          xmlns:util="http://www.apache.org/1999/XSP/Util">
  <ventes>
    <util:include-uri href="http://www.foo.com/ventes.jsp?rayon=charcuterie"/>
    <util:include-uri href="http://www.foo.com/ventes.jsp?rayon=fromages"/>
    <util:include-uri href="http://www.foo.com/ventes.jsp?rayon=primeurs"/>
  </ventes>
</xsp:page>
```

Technologie émergente, à la mode
Repose entièrement sur XML

Permet de réaliser l'intégration de services sur internet

Middleware basé sur une architecture XML
→ Les échanges se réalisent en XML avec le protocole SOAP

Intégration d'applications inter-entreprises

Services Web { Publication
Localisation
Utilisation

Model : logique d'accès aux données

View : page JSP qui assemble en HTML le résultat

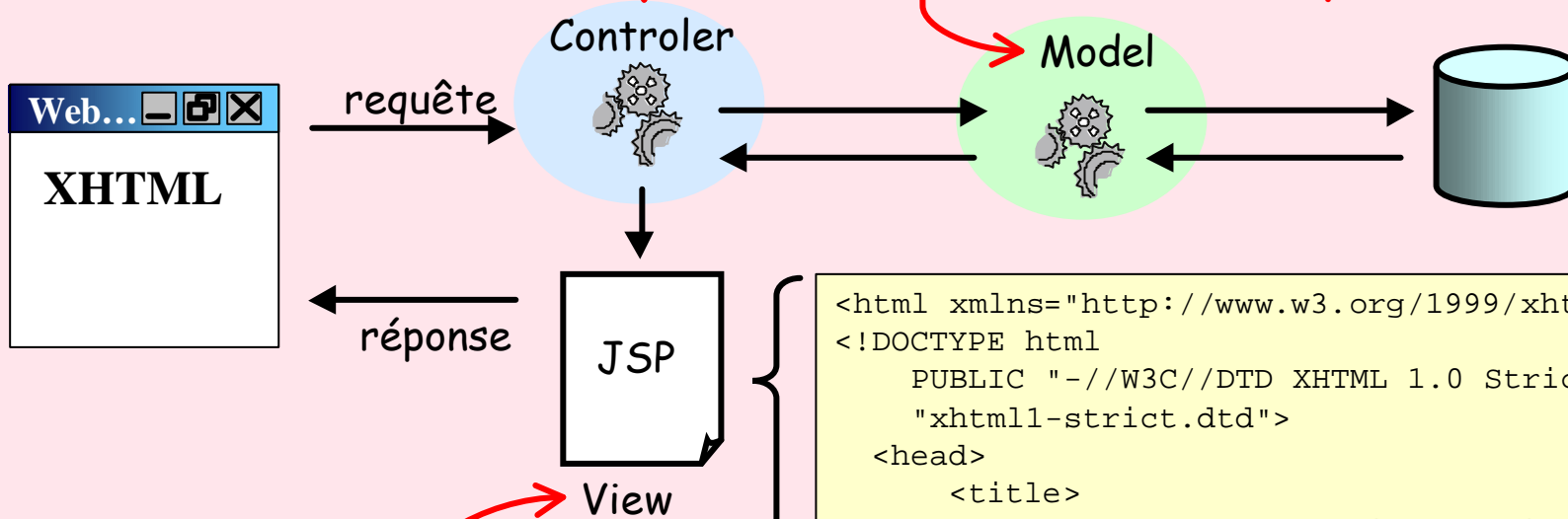
Controler : servlet qui interprète les requêtes du client

- Décode les requêtes
- Récupère les données brutes
- Applique les transformations XSLT

Développeur Web

- Se connecte aux sources de données
- Accède aux données

Programmeur du Système d'Information



- Récupère les données calculées
- Les insère dans la page HTML

Web designer

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "xhtml1-strict.dtd">
<head>
  <title>
    <%=request.getAttribute("title")%>
  </title>
  <%=request.getAttribute("meta-datas")%>
</head>
<body background="fond.jpg">
  
  <%=request.getAttribute("toolbar")%>
  <%=request.getAttribute("content")%>
</body>
</html>
```

Serveur configuration

Servlet

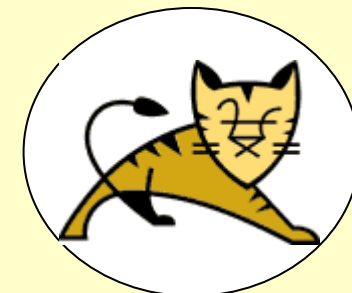
Descripteur d'applications Web

```

<?xml version="1.0"?>
<Server port="8005" shutdown="SHUTDOWN" debug="0">
  <!-- Enable JMX MBeans
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" debug="0"/>
  <!-- ===== Internal Servlet Engine
  <Service name="Catalina">
    <!-- Non-SSL Connector
    <Connector className="org.apache.catalina.connector.HTTPS"
      port="8443"
      enabled="true"
      acceptCount="10"
      scheme="https"
      <!-- Internal Servlet Engine
    <Engine name="Catalina" debug="0">
      <!-- Internal Servlet Engine
      <Logger className="org.apache.catalina.logger.FileLogger"
        directory="logs"
        prefix="catalina."
      <!-- Internal Servlet Engine
      <Realm className="org.apache.catalina.realm.RealmBase">
      <!-- Services Virtual Hosts
      <Host name="localhost"
        unpackWARs="true"
        <!-- Insert Context
      </Host>
    </Engine>
  </Service>
  <!-- ===== Web Services
  <Service name="WebServices">
    <!-- Non-SSL Connector
    <Connector className="org.apache.catalina.connector.HTTPS"
      port="8443"
      enableLookups="true" redirectPort="8443"
  
```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>
      org.apache.catalina.servlets.DefaultServlet
    </servlet-class>
    <init-param>
      <param-name>listings</param-name>
      <param-value>true</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <mime-mapping>
    <extension>html</extension>
    <mime-type>text/html</mime-type>
  </mime-mapping>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
  
```

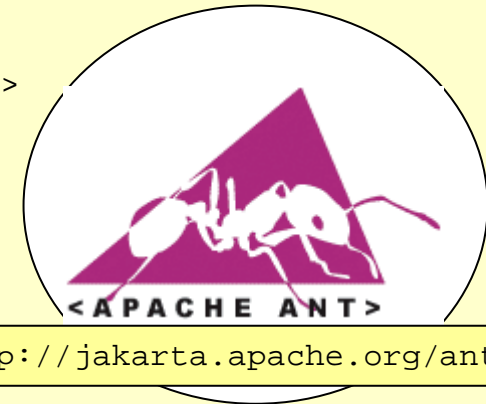


```

<!-- Setting up the example to prepare to build and deploy -->
<project name="doc" default="" basedir=".">
  <target name="init">
    <tstamp/>
  </target>
  <!-- This section sets properties used in rest of build file -->
  <property name="build" value="build" />
  <property name="src" value="src" />
  <property environment="myenv" />
  <!-- This section takes care of setting up your CLASSPATH -->
  <path id="classpath">
    <fileset dir="${myenv.JWSDP_HOME}/common/lib">
      <include name="*.jar"/>
    </fileset>
  </path>
  <!-- This section prepare the directory structure needed for Web applications -->
  <target name="prepare" depends="init" description="Create build directories.">
    <mkdir dir="${build}/WEB-INF/classes" />
    <copy todir="${build}">
      <fileset dir="${src}" excludes="**/*.java,**/*.old"/>
    </copy>
  </target>
  <!-- This section compiles the Java files and copies the HTML and JSP pages to the appropriate directory -->
  <target name="build" depends="prepare" description="Compile app Java files and copy HTML and JSP pages to the appropriate directory">
    <javac srcdir="${src}" destdir="${build}/WEB-INF/classes">
      <include name="**/*.java" />
      <classpath refid="classpath"/>
    </javac>
    <copy todir="${build}">
      <fileset dir=".">
        <include name="*.html" />
        <include name="*.jsp" />
      </fileset>
    </copy>
  </target>

```

Ant


<http://jakarta.apache.org/ant>

ANT : Automate de
déploiement d'applications
Java
(make file pour Java)

Architecture J2EE

Descripteurs EJB : fichiers XML

