



# PHP 5 et XML

Éric Daspet  
Novembre 2004



# Les points abordés

---

- ▶ Pourquoi le XML ?
- ▶ Le point avec PHP 4
- ▶ Un outil simple, SimpleXML
- ▶ Un outil complet, DOM
- ▶ Gestion avancée avec XSLT
- ▶ SOAP et Web Services



# Qu'est-ce ?

---

- description de vos données à l'aide de balises

```
<bibliotheque>
```

```
  <livre isbn="2-212-11323-4">
```

```
    <titre> PHP 5 avancé </titre>
```

```
    <editeur> Eyrolles </editeur>
```

```
  </livre>
```

```
<livre> ... </livre>
```

```
<livre> ... </livre>
```

```
</bibliotheque>
```

- mêmes héritages que le HTML



# Pourquoi XML ?

---

- Pérennité : format standardisé
  - normalisé par le W3C
- Universalité : utilisable pour vos données
  - adaptable et peu spécifique
- Interopérabilité : présent partout
  - aussi dans vos autres applications et langages



# Où ?

---

- Les logiciels de bureautique
  - MS Office, Open Office
- Le Web
  - XHTML, RSS, P3P, SVG, MathML
- Les applications métier
  - SOAP et Web Services, XmlPay



# Avec PHP 4

---

- Pour manipuler le XML
  - Extensions SAX et DOMXML
  - Classes spécialisées via PEAR
- Pour converser
  - Extension XML-RPC, classe PEAR SOAP
- Pour transformer
  - Extension DOMXSLT et Sablot



# Le problème de PHP 4

---

- **Manque de simplicité**
- **Pas de « bonne » solution**
  - Pour le XSLT, Sablot est lent
  - DOMXML n'est pas fini et pas standard
  - Les classes PEAR compensent à peine
- **XML est présent, mais peut mieux faire**





# Simple XML



# SimpleXML : qu'est-ce ?

---

- Extension PHP 5 pour lire du XML

```
$biblio = simplexml_load_file('biblio.xml');  
foreach($biblio->livre as $livre) {  
    echo $livre->titre ;  
    echo ' aux éditions ' ;  
    echo $livre->editeur ;  
}
```



# SimpleXML : textes

---

- Lire le contenu textuel d'un noeud
  - Transtypage vers une chaîne de caractères  
`$texte = (string) $noeud ;`
  - Géré automatiquement dans certains cas  
`echo $noeud ; $texte = "$noeud" ;`
- Lecture de `<a>1<b>2</b>3</a>`
  - Impossible de tout remettre dans l'ordre



# SimpleXML : espaces de nom

---

- Espaces de noms ignorés par défaut
- Utiliser les espaces de noms
  - `$ns = $noeud->children('uri') ; $ns->nomBalise`
  - `$att = $neoud->attributes('uri') ; $att['nomAttr']`
- Impossible d'avoir les balises de manière ordonnée avec leur espaces de nom



# SimpleXML : avantages

---

- Aucune connaissance requise en XML
- **Simple à manipuler / utiliser**
- Extension native donc rapide
- Compilée par défaut (donc portable)
- Entièrement orientée objet
  - Profite des fonctionnalités PHP 5



# SimpleXML : défauts

---

- Peu de détail sur ce qu'on manipule
- Récupération complexe des noeuds texte
- Utilisation des espaces de noms imparfaite
- Lecture seule (ou presque)



# SimpleXML : pour quoi ?

---

- XML type fichiers de configuration
  - Pas trop gros (quelques Mo tout de même)
  - Peu complexes
  - Très hiérarchiques
  - En lecture
- Pour accéder très rapidement à un élément





# DOM avec PHP 5



# DOM : qu'est-ce ?

---

- Une interface de programmation standardisée
- Identique dans tous les langages ou presque
- Manipulation en détail de la syntaxe XML
- Entièrement Orienté Objet
- En lecture, en création et en modification



# DOM : PHP 5

---

- Nouvelle extension, **complète et standard**
- Utilise les fonctionnalités PHP 5 (wrappers)
- Compilée et activée par défaut (**portable**)
- Basée sur la libxml2 (projet gnome)
- Gestion de Xinclude



# DOM : exemple

---

```
$doc = DomDocument::load('biblio.xml') ;
$biblio = $doc->documentElement ;
foreach($biblio->childNodes as $livre) {
    if ($livre->nodeType == XML_ELEMENT_NODE) {
        foreach($livre->childNodes as $node) {
            if ($node->nodeType == XML_ELEMENT_NODE
                && $node->tagName == 'titre' ) {
                foreach($node->childNodes as $txtNode) {
                    echo $txtNode->nodeValue ;
                }
            }
        }
    }
}
```



# DOM : validation

---

- Vérifier la validité de vos documents
  - par DTD (W3C, peu strict et peu complexe)
    - `$doc->validate()` ;
  - XML Schema (W3C, strict mais complexe)
    - `$doc->schemaValidate()` ;
  - RelaxNG (ISO, strict et peu complexe, récent)
    - `$doc->relaxNGValidate()` ;



# DOM : pas que des avantages

---

- Beaucoup trop verbeux
  - parfois long pour faire des choses simples
- Peu intuitif
  - Surtout pour les noeuds texte
- Peu simple pour joindre un noeud précis



# Simple XML et DOM

---

- Cumuler les avantages ?
- Passer de l'un à l'autre au besoin :
  - `$domObj = DOM_import_SimpleXML( $sxObj )`
  - `$sxObj = SimpleXML_import_DOM( $domObj )`
- Opération peu coûteuse (même backend)



# XPATH

---

- Xpath :sélection de fragments XML
- Intégré à SimpleXML et à DOM

```
$xml = simplexml_load_file('biblio.xml');  
$meslivres = '/bibliotheque/livre[@isbn]';  
$liste = $xml->xpath($meslivres);  
foreach ($liste as $livre) {  
    echo $livre['isbn'];  
}
```

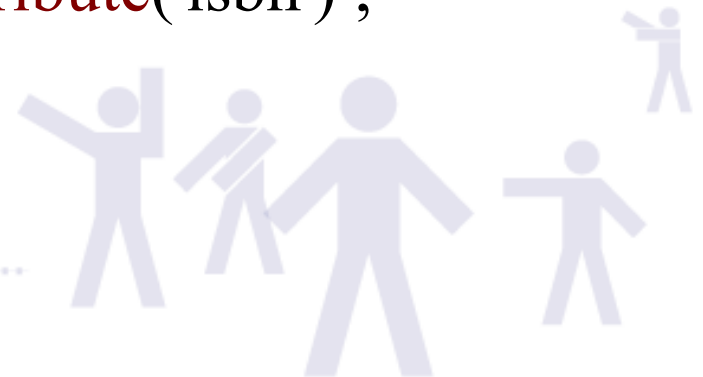


# XPATH

---

- Le même avec DOM

```
$doc = new DOMDocument::load('biblio.xml');  
$xpath = new DOMXPath($doc);  
$meslivres = '/bibliotheque/livre[@isbn]';  
$liste = $xpath->query( $meslivres );  
for ($i=0 ; $i<$liste->length; $i++) {  
    echo $liste->item($i)->getAttribute('isbn');  
}
```



# Autres possibilités PHP 5

---

- Lecture de XML mal formé
  - via la librairie Tidy
  - via XML\_HTMLSax dans PEAR
  - via loadHTML() dans DOM
- Possibilité de spécialiser les classes PHP
  - dans simpleXML et dans DOM
  - rajouter des méthodes, faire un objet XML...



# Implémentations économes

---

- DOM et SimpleXML sont lourd
  - tout le doc est en mémoire (taille x3)
- Implémentations qui lisent par partie
  - SAX (complexe)
  - XML\_Reader (dans PECL)
- Défauts :
  - Impossible de revenir en arrière
  - Absence de contexte
  - Tout est géré à la main





# Gestion XSLT



# XSLT : qu'est-ce ?

---

- Un langage XML
- Transforme un fichier XML en un autre
  - MS Word -> OpenOffice
  - Sorties en RSS, XHTML, ...
- N'est pas limité à une sortie XML
  - OpenOffice -> PDF



# XSLT : PHP 4

---

- Une extension avec Sablotron
  - lente
  - pas de gestion de EXSLT
- Une extension avec libxslt
  - basée sur domxml qui n'était pas finie
  - donc peu présente sur les hébergements



# XSLT : PHP 5

---

- Une nouvelle extension basée sur libxslt
  - Intégrée à l'extension DOM
  - **Performante**

```
$xml = DomDocument::load('example.xml');  
$xsl = DomDocument::load('example.xsl');  
$proc = new xsltprocessor;  
$proc->importStyleSheet($xsl);  
echo $proc->transformToXML($xml);
```



# XSLT : goodies

---

- Utilisation de fonctions PHP dans XSLT

```
<xsl:value xmlns:php="http://php.net/xsl"  
  select="php:functionString('gettext', 'bonjour')" />
```

```
$proc = new xsltprocessor;  
$proc->registerPhpFunctions();
```



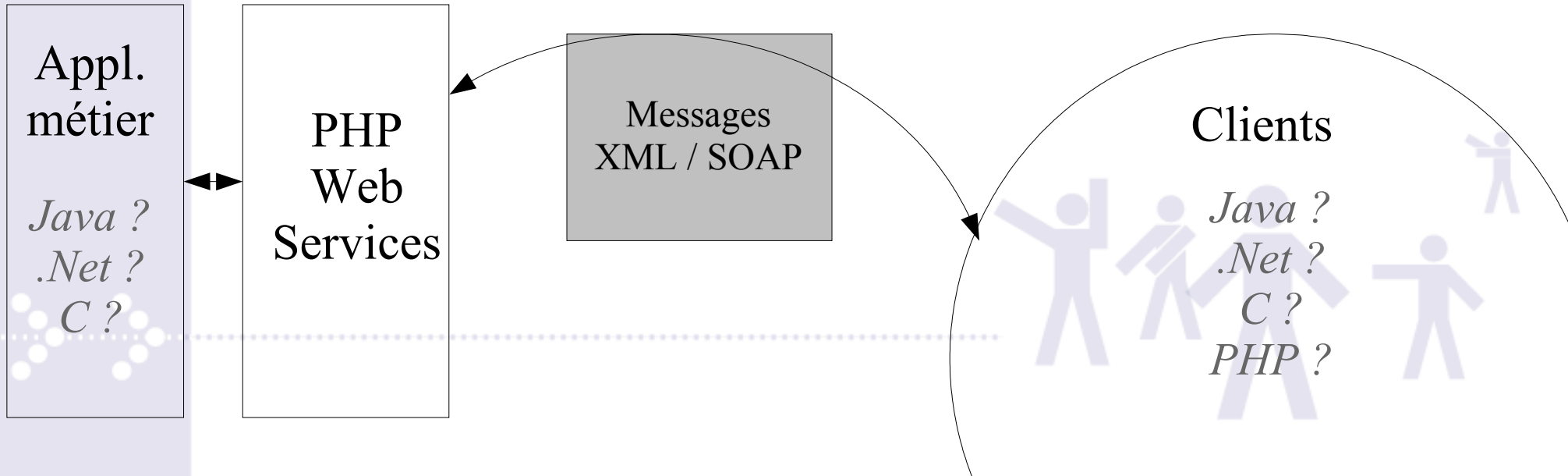


# Services Web



# Web Services : qu'est-ce ?

- L'appel d'une méthode ou d'un service distant
- Indépendance face aux langages
- Philosophie Orientée Objet



# Web Services : PHP 5

---

- Une nouvelle extension SOAP
  - toujours marquée comme « non finie »
  - mais relativement complète et stable
- Un point fort : **la simplicité**



# Web Services : client

---

- On utilise un objet distant et ses méthodes comme s'il était en local :

```
$wsdl = 'http://provider/service/descr.wsdl' ;
```

```
$client = new SoapClient( $wsdl );
```

```
list($val1, $val2) =
```

```
    $client->fonctionDistante($param1, $param2) ;
```



# Web Services : serveur

---

- On définit une classe locale et tout le reste est géré par l'extension :

```
class monServeur {  
    function maMethodeSoap1($nom) { return "hello $nom" ; }  
    function maMethodeSoap2() { ... }  
}  
$serveur = new SoapServer('fichier.wsdl');  
$serveur->setClass('monServeur');  
$serveur->setPersistence(SOAP_PERSISTENCE_SESSION);  
$serveur->handle() ;
```



# Web Services : autres

---

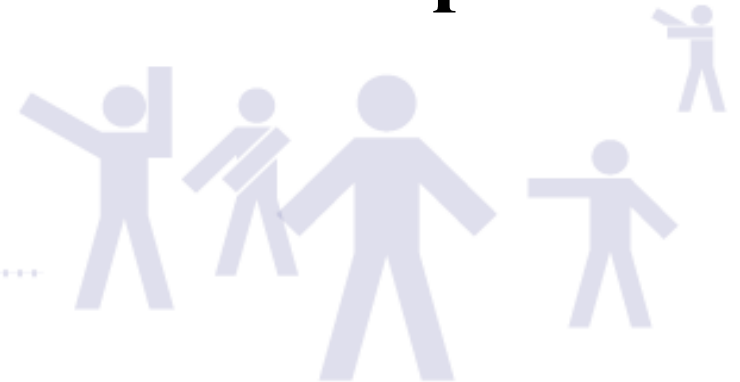
- Des classes PEAR spécialisées
  - Moteur de recherche (Google)
  - Édition (Amazon)
  - Enchères, vente (Ebay)
  - Météo
  - Cours de la bourse





# XML et PHP 5

- Des outils **simples**, très simples à utiliser
- Des outils **complets**, plus détaillés
- Des outils **adaptés pour tous et pour tout**





# Questions ?



**Eric Daspét**

Forum PHP AFUP  
Novembre 2004

