

La mise en ligne des bases de données sur internet

les techniques d'extraction, les formulaires
et intégration dans un site WEB

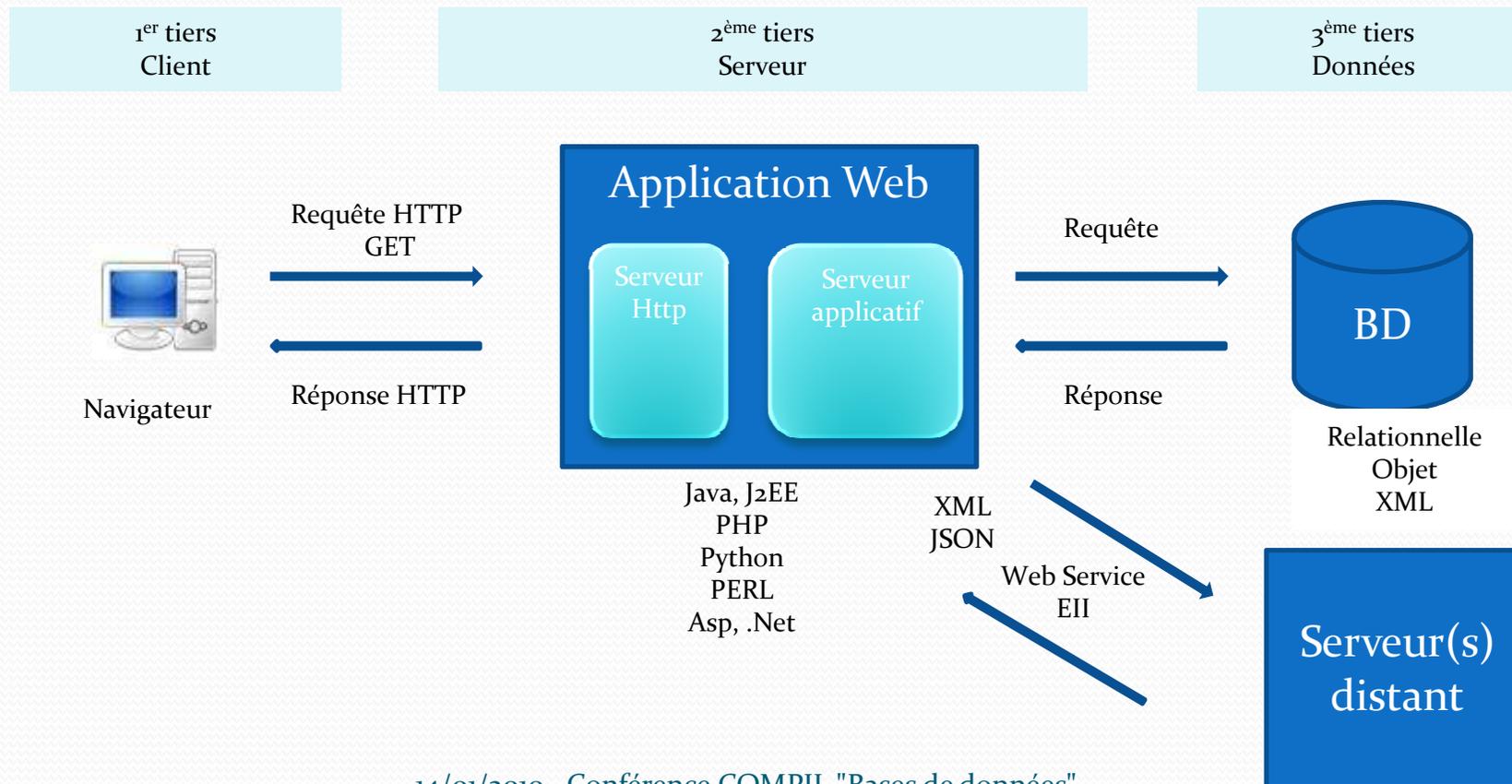
Edyta Bellouni
MSHS-T, UMS838

Plan

- L'extraction des données pour un site en ligne
 - Architecture et techniques
 - Les différents formats d'affichage
 - Représentation des données sous forme de graphiques
- Les formulaires de saisie
- La sécurité des données
- Mise en ligne d'une BD MySQL avec PHP
 - Avantages et inconvénients de ce procédé
- Intégration d'une BD dans un CMS
 - Fonctionnement d'un CMS
 - Exemple de SPIP
 - Avantages et inconvénients de ce procédé
- Utilisation d'un framework Web
 - Pourquoi utiliser un framework ?
 - Comment choisir son framework ?
 - CMS ou framework ?

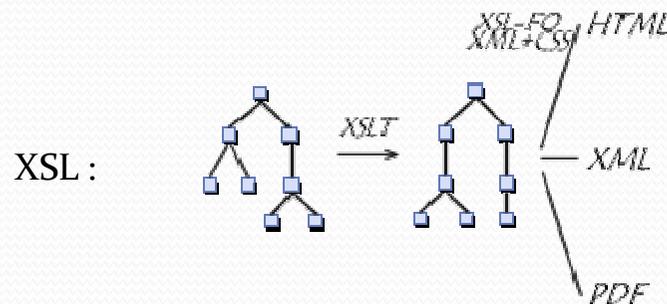
L'extraction des données pour un site en ligne

- Architecture 3 tiers
- Site Web dynamique : les pages sont générées sur le serveur



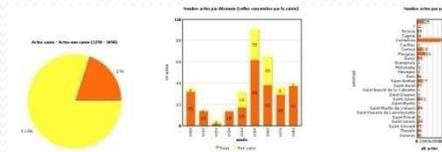
L'extraction des données pour un site en ligne

- Les différents formats d'affichage
 - HTML
 - XHTML : permet l'utilisation de nouvelles balises grâce aux espaces de noms, ex SVG
 - WML (Wireless Markup Language) : langage à balises pour les téléphones portables
 - PDF
 - XML
- Les données XML
 - Les fichiers XML peuvent être transformés à l'aide des feuilles de style XSL (recommandation W₃C)



L'extraction des données pour un site en ligne

- Les principaux formats graphiques pour le Web
 - Formats vectoriels : SVG, VML
 - Format XML
 - Google Maps utilise VML pour le rendu vectoriel avec IE 5,5+ et SVG pour les navigateurs le supportant
 - L'avantage du vectoriel
 - Le référencement
 - Formats open source et standards
 - SVG est une recommandation du W3C
 - Flash
 - Représentation binaire : fichiers SWF
 - L'avantage du SWF : meilleure portabilité grâce au lecteur Flash bien implanté
 - Mais format partiellement propriétaire
- Quelques outils et bibliothèques pour générer des graphiques à partir des données stockées en base
 - SVG / VML
 - Raphaël : Librairie JavaScript
 - Dojo gfx : API pour graphiques vectoriels
 - Flash
 - Flex
 - Open Flash Charts (Ruby, .Net, Perl, Java, Python, ColdFusion, GWT, SmallTalk, Penttaho, C) : libre et open source : <http://teethgrinder.co.uk/open-flash-chart-2/downloads.php>
 - Fusion Charts (PHP, ASP.NET, JSP, ColdFusion, Python, RoR) : existe en version libre et payante <http://www.fusioncharts.com/free/>



La mise à jour des données => les formulaires

- Les formulaires sont l'interface des internautes pour interagir avec notre application

Nom *	<input type="text"/>	Prénom *	<input type="text"/>
Statut	<input type="text"/>		
Adresse personnelle *	<input type="text"/>		
Code Postal *	<input type="text"/>	Ville *	<input type="text"/>
Pays *	France <input type="button" value="v"/>		
Adresse mail *	<input type="text"/>		
Téléphone	<input type="text"/>		
Organisme / Etablissement	<input type="text"/>		



La sécurité des données

- Veiller à l'intégrité des données
 - Contrôle des formulaires côté serveur, en plus du contrôle côté client
 - Eviter la réexécution d'un POST
 - Utiliser une redirection pour afficher la page de retour
- Attaques par injection de code SQL
 - Utiliser des procédures stockées ou des requêtes SQL préparées
 - Utiliser des comptes utilisateurs SQL avec des droits restreints (en lecture seule si possible)
 - Filtrer des mots clés comme EXEC, SELECT, INSERT etc...
 - Echapper les caractères spéciaux comme ' , « , \ul> - PHP offrait jusqu'à présent un paramètre « magic_quote » qu'il suffisait d'activer pour échapper les caractères spéciaux, mais depuis PHP5 cette fonctionnalité est dépréciée et sera supprimée en PHP6 !
 - => Il faut utiliser la méthode mysql_real_escape_string

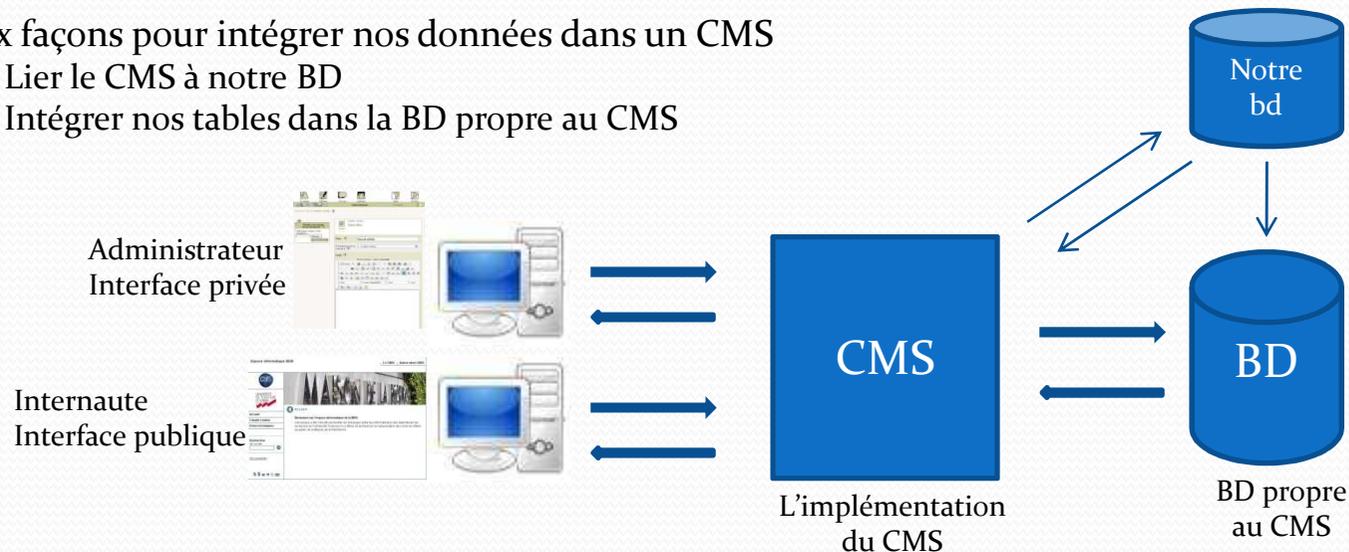
Mise en ligne d'une BD MySQL avec PHP

- PHP : un langage créé pour des applications Web
- Avantages
 - Rapide à mettre en place pour une application simple
 - EasyPHP : un environnement de développement
 - Un serveur HTTP : Apache
 - Un langage de programmation : PHP
 - Une base de données : MySQL (ou SQLite)
 - Une interface d'administration pour la base de données : phpMyAdmin ou SQLiteManager
 - Créer un alias pour chaque site Web
 - Les environnements de déploiement de type LAMP (Linux, Apache, MySQL, PHP) sont très répandus
- Inconvénients
 - Fastidieux pour un site Web complet dès que l'on souhaite intégrer une gestion de droits, un habillage plus travaillé, un forum etc ...
 - On réinvente la roue
 - On risque de produire du code « sale » : difficile à maintenir

Intégration d'une BD dans un CMS

=> fonctionnement d'un CMS

- CMS : Content Management System : système de gestion de contenu
Logiciel destiné à la conception et à la mise à jour dynamique de sites web (sans programmation)
 - L'interface privée permet d'administrer et de structurer le site
 - Souvent nombreuses fonctionnalités de base : forums, pages à accès restreint, recherche sur le site
 - Possibilité de rajouter des plugins
- Quelques CMS du marché (Open Source)
 - CMS Made Simple, Spip, Drupal, Joomla
- Deux façons pour intégrer nos données dans un CMS
 - Lier le CMS à notre BD
 - Intégrer nos tables dans la BD propre au CMS



Intégration d'une BD dans un CMS

=> exemple de SPIP

- La base de données de SPIP
 - Serveurs SQL supportés actuellement : MySQL, PostgreSQL, SQLite
 - ~40 tables, plus si ajout de plugins
- Intégration de données externes
 - Accès à une base de données externe
 - SPIP 2.0 : http://www.spip.net/fr_article3681.html uniquement en lecture actuellement, la possibilité de mise à jour est à l'étude
 - Avant SPIP 2.0 : possible, mais via le code PHP
 - Ajout de nouvelles tables dans la base de données de SPIP (préconisé)
 - Via une console d'administration (phpMyAdmin, ...)
 - Via le plugin TableDATA
 - Outil d'administration de la base de données dans l'interface privée de SPIP (une sorte de phpMyAdmin pour les « non informaticiens »)
 - Existe pour les versions 1.9.1, 1.9.2 et 2.0 en version stable
- Plugin pour la création de formulaires de saisie
 - Plugin Forms & Tables
 - Création des formulaires de saisie et des tables associées
 - Collecte des données
 - Export CSV
 - Version de test pour SPIP 2

Nom * Prénom *

Statut

Adresse personnelle *

Code Postal * Ville *

Pays *

Adresse mail *

Téléphone

Organisme / Etablissement

Intégration d'une BD dans un CMS

=> exemple de SPIP

- Afficher le contenu des tables spécifiques dans les pages publiques
 - Créer ou modifier des squelettes SPIP
 - Utiliser le langage SPIP
 - Intégrer du code PHP
- Format des tables à respecter si on utilise le langage SPIP pour l'extraction des données
 - Noms des tables et des colonnes en minuscule en base et en majuscules dans le langage SPIP
 - Mêmes noms pour les colonnes liées par une contrainte de clé étrangère si on souhaite utiliser des jointures dans le langage SPIP

```
<INCLUDE(fond=entete) (type_page=auteur) (id_auteur) (lang)>

<BOUCLE_auteur_principal(AUTEURS) (id_auteur)>

[ (#LOGO_AUTEUR|image_reduire(200,200))]
<h2>#NOM</h2>
[<p class="soustitre"><:voir_en_ligne:>&nbsp;&nbsp;&nbsp;: <a href="#URL_SITE">
[ (#NOM_SITE|sinon([ (#URL_SITE|couper(80))]))</a></p>]

[<p> (#BIO)</p>]

[ (#FORMULAIRE_ECRIRE_AUTEUR)]

<B_articles>
#ANCRE_PAGINATION
<h3 class="spip"><:articles_auteur:></h3>
[<p> (#PAGINATION)</p>]
<ul class="liste-liens">
  <BOUCLE_articles(ARTICLES) (id_auteur) (par date){inverse} (pagination 10)>
  <li><a href="#URL_ARTICLE"> [ (#TITRE|supprimer_numero)]</a>&nbsp;&nbsp;&nbsp;
  <span class="petit" style="color:#999999;"> [ (#DATE|affiche)]</span></li>
  </BOUCLE_articles>
</ul>
</B_articles>

[<h3><:info_notes:></p><p> (#NOTES)</h3>]

</BOUCLE_auteur_principal>

<INCLUDE(fond=pied) (lang)>
```

```
<BOUCLE_chercheur_principal(CHERCHEUR){id_cher}>
<div class="groupe titre">
<h2>#PRENOM_CHER #NOM_CHER</h2>
</div>
</BOUCLE_chercheur_principal>
```

```
<?php
$tableau = get_interv_non_commun(#ID_CHER);
if(isset($tableau)) {
  echo "<i> Spécifiques : </i>";
  echo "<ul>";
  foreach($tableau as $value) {
    echo "<li>".$value["descr_interv"]."<br/></li>";
  }
  echo "</ul>";
}
?>
```

Intégration d'une BD dans un CMS

- Avantages
 - Permet de profiter d'un habillage et des fonctionnalités existants
 - Rapide à mettre en œuvre une fois passé l'étape d'apprentissage ou si on utilise un plugin existant qui correspond à notre besoin
 - S'adapte bien à un besoin de création d'un site Web complet où on souhaite intégrer des données supplémentaires
- Inconvénients dans le cas de modification du code
 - Réutilisation d'une architecture qui au départ n'était pas prévue pour ça
 - La documentation n'est pas orientée vers cette utilisation du CMS
 - La prise en main est souvent difficile, il faut comprendre la logique de développement du CMS en question
 - Peut donner du code « sale » ou redondant
 - En cas de nouvelles versions du CMS ou de patches de sécurité, il faut répercuter ses modifications à la main vers la nouvelle version
- Inconvénients dans le cas de l'utilisation de plugins
 - L'évolution d'un plugin dépend de son créateur et pas forcément de la communauté
 - Les fonctionnalités peuvent être limitées ou trop génériques et ne pas couvrir entièrement notre besoin, auquel cas on risque d'être amené à retoucher le code

Utilisation d'un framework Web

- Pourquoi utiliser un framework
 - Offre un cadre de développement pour créer des sites Web dynamiques maintenables et sécurisés et respectant les patterns de développement
 - Permet d'alléger la charge de développement liée aux fonctionnalités récurrentes
- En général un framework Web fournit :
 - Une librairie d'accès aux données ou ORM (Object Relational Mapping)
 - Des templates HTML
 - Une gestion du cache
 - Une gestion de l'authentification et des autorisations sécurisés
 - Un pattern MVC
- Quelques frameworks Web du marché :
 - Java : Spring, Apache Struts
 - PHP : Symfony, CakePHP, Zend Framework
 - Perl : Catalyst
 - Python : Django
 - Ruby : Ruby On Rails
 - TCL : OpenACS
- Comment choisir son framework ?
 - Utiliser le langage avec lequel vous êtes à l'aise
 - Choisir en fonction des fonctionnalités offertes par rapport à votre besoin
 - Communauté et documentation

Utilisation d'un framework Web

- Liste et comparatif des frameworks Web sur Wikipédia : http://en.wikipedia.org/wiki/List_of_web_application_frameworks
 - Exemple pour les frameworks ASP.NET

Project	Ajax	MVC framework	MVC Push/Pull	i18n & l10n?	ORM	Testing framework(s)	DB migration framework(s)	Security Framework(s)	Template Framework(s)	Caching Framework(s)	Form Validation Framework(s)
ASP.NET MVC	Yes	Yes	Push		ORM-independent	Unit Tests		ASP.NET Forms Auth	pluggable (default is WebForms)	Yes	Yes (client-side via plugins)
BFC	Yes	Not mandatory	Push & Pull	Yes	through active data dictionary	Unit Tests	SQL Server, Oracle, DB2, Sybase, MySQL	security groups and rules	Yes	metadata and result sets	data dictionary-driven
DotNetNuke	Yes	No	Pull	Yes	SubSonic, NHibernate	Unit Tests	Yes	ACL-based (OpenID, LiveID, Active Directory, LDAP, CardSpace, ASP.NET Forms Auth)	Yes	Pluggable	ASP.NET Validators, built-in API
MonoRail	Prototype	Active record pattern	Push	Yes	Active record pattern	Unit Tests		via ASP.NET Forms Authentication	Yes	Yes	Yes
Project	Ajax	MVC framework	MVC Push/Pull	i18n & l10n?	ORM	Testing framework(s)	DB migration framework(s)	Security Framework(s)	Template Framework(s)	Caching Framework(s)	Form Validation Framework(s)

- Les notions de framework, CMF (Content Management Framework : plateforme de gestion de contenu) et CMS s'entrecroisent
 - Les CMS ont beaucoup évolué ces dernières années et concurrencent les frameworks, ex Drupal
 - Liste de CMS sur Wikipédia : http://en.wikipedia.org/wiki/List_of_content_management_systems
 - Liste de CMF sur Wikipédia : http://en.wikipedia.org/wiki/Content_management_framework