



# Mozilla et XUL

## Interfaces riches dans les applications web

Par Laurent Jouanneau  
<http://xulfr.org>

Libr'east 2004  
<http://www.idile.org/libreast/>

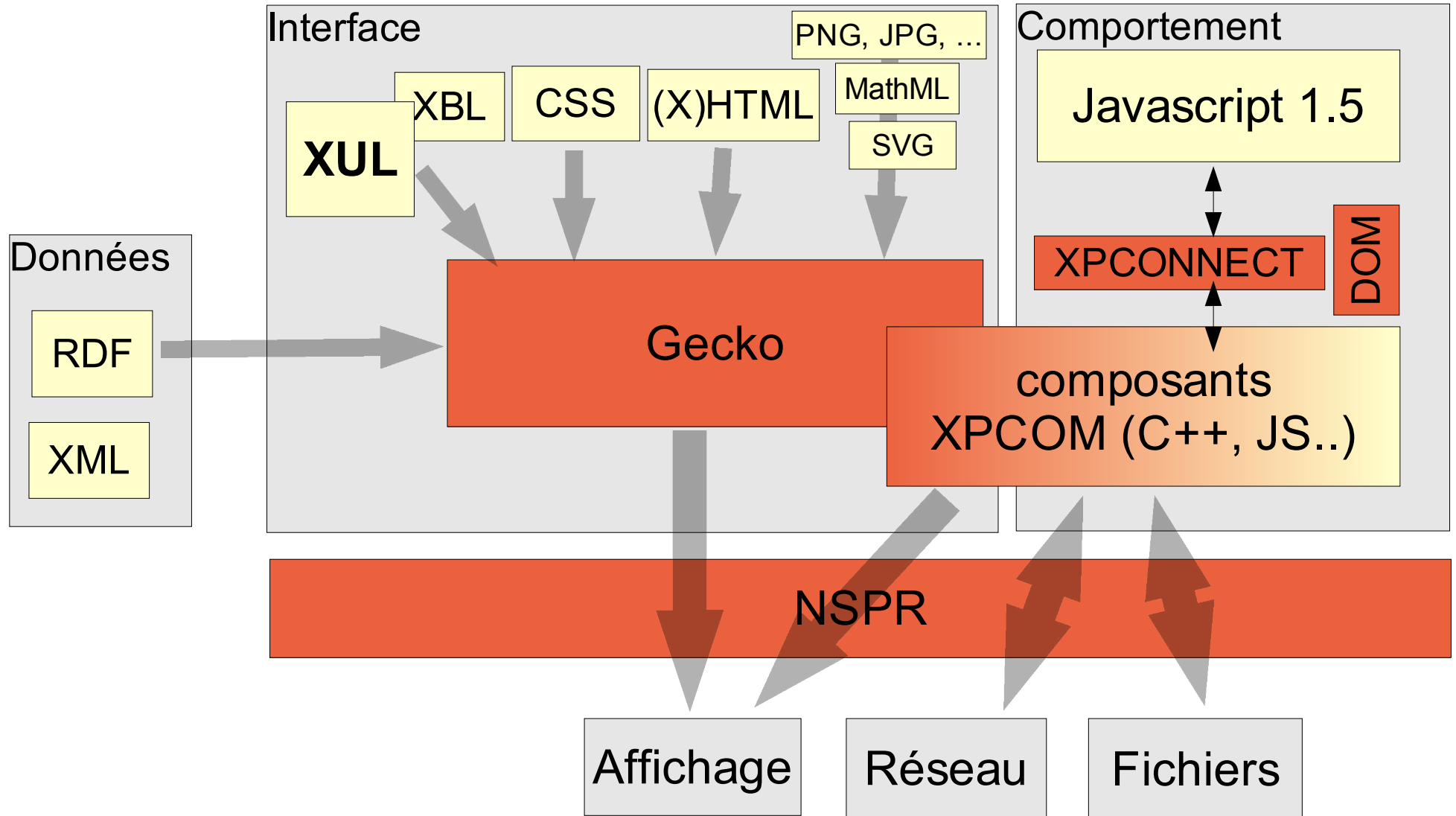


- Présentation de Mozilla
- Les technologies incluses dans Mozilla
- Utiliser les technologies de Mozilla pour des applications Web et desktop

## Présentation de Mozilla

- Suite Internet : navigateur, client mail...
- Nom de code du projet pour développer le navigateur Netscape
- Courant 1998 : libération du code source NS4 et naissance de Mozilla.
- 5 Juin 2002 : version 1.0
- Problématique : faire un navigateur multi-plateforme sans "exploser" les temps de développement

- Mozilla n'est pas seulement un navigateur
- Mozilla est un framework pour développer des applications desktop et web
- Mozilla contient un ensemble de technologies permettant de réaliser des applications multi-plateforme et reposant sur bon nombre de standards (du W3C et ECMA).



- Plateforme mozilla
- Éléments de l'application

## Les technologies incluses dans Mozilla

- XML-based User interface Language
- Format xml, permettant de décrire une interface utilisateur
- Contient un ensemble de balises correspondant à des composants graphiques courants
  - boutons, case à cocher, champs de saisie
  - boîte de liste, liste en arbre, grilles
  - barres de menus, menu déroulant, menu popups
  - barres d'outils, barre de statut, onglets etc..
  - zones d'edition avancée, afficheur de page web..
  - etc...



## exemple1.xul :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window title="Recherche de fichiers"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <description>Entrez votre critère de recherche ci dessous et
  appuyez sur le bouton Rechercher.</description>

  <groupbox orient="horizontal">
    <caption label="Critère de recherche"/>

    <menulist id="searchtype">
      <menupopup>
        <menuitem label="Nom"/>
        <menuitem label="Taille"/>
        <menuitem label="Date"/>
      </menupopup>
    </menulist>
    <textbox id="find-text" flex="1" style="min-width: 15em;"/>
  </groupbox>

  <button id="find-button" label="Rechercher" default="true"/>
</window>
```

Voir "table périodique" de xul (<http://www.hevanet.com/acorbin/xul/top.xul> )

- Modifiable dynamiquement via DOM javascript
- Jeu de balise extensible avec XBL
- Système d'overlay
  - permet de répartir la déclaration dans plusieurs fichiers
  - permet de modifier une interface déjà déclarée sans toucher à ses sources
- Attributs pour répondre aux évènements (onclick, ondragdrop, onkeypress, onoverflow ...)
- Localisable (utilisation de DTD + entités)

- eXtensible Bindings Language
- Note du W3C
- Permet de définir un composant d'interface :
  - son contenu XUL / XHTML / autre
  - son implémentation
    - propriétés et méthodes que l'on pourra appeler en javascript
    - attributs de la balise qui le représentera
    - constructeur/destructeur
    - les traitements en réponse à des évènements
- possibilités d'héritage
- Portable : Indépendant de l'environnement (OS / GUI)

Exemple : créer un nouveau composant, représenté par une nouvelle balise `<slideshow>`, permettant d'afficher un à un d'autres composants, par des boutons suivant / précédent

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<?xml-stylesheet href="styles.css" type="text/css"?>
```

```
<window
```

```
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
```

```
<slideshow previoustext="Précédent" nexttext="Suivant" flex="1">
```

```
<button label="Bouton 1"/>
```

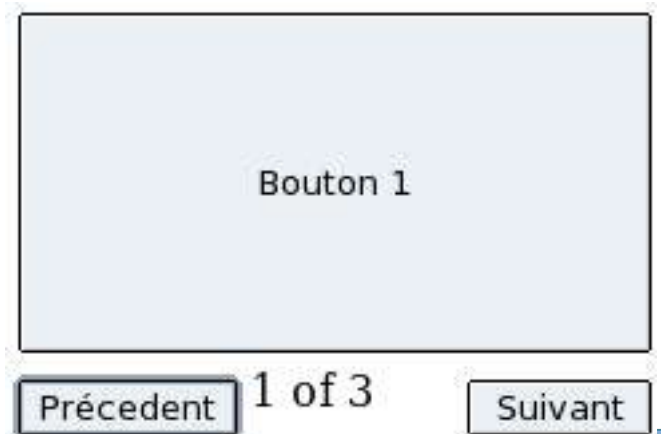
```
<checkbox label="Checkbox 2"/>
```

```
<textbox/>
```

```
</slideshow>
```

```
</window>
```

[Voir source slideshow.xbl](#)



- Langage permettant de spécifier le rendu d'un fichier XML à l'écran
- Connue pour son utilisation dans les sites HTML
- Utilisée dans Mozilla pour "décorer" l'interface
- Utilisée notamment pour les thèmes graphiques
  - ex : couleur, bordure des boutons, des boîtes en général, apparence des menus etc..
- Respecte en grande partie la spécification CSS 2 éditée par le W3C, et implémente quelques styles du futur CSS 3

- Cross-platform Component Object Model
- Comparable à la technologie COM de MS
- Permet de définir des composants réutilisables
- une grande partie du framework Mozilla reposent sur des composants XPCOM
- Programmé en C++, Javascript, ou Python (perl ou autre à l'avenir)
- Chaque XPCOM offre une/des interfaces (méthodes et propriétés)
  - pour être utilisable dynamiquement (pour un programme voulant l'utiliser, pas besoin de recompilation / liaison)
  - pour avoir plusieurs implémentations selon la plateforme

- Des composants pour
  - implémenter les éléments XUL de base ou autres éléments graphiques (html, svg...)
  - offrir des services
    - son
    - réseau
    - gérer le drag'n drop
    - accéder aux fichiers
  - faire des traitements fonctionnels
    - gérer les signets
    - gérer les cookies
    - lire des fichiers RDF
  - etc..

- Resource Description Framework
- Standard W3C
- Format XML permettant de stocker des données de façon "relationnel"
- Utilisé pour stocker des données du navigateur (liste sidebars, downloads, thèmes) ou du client mail (les mails, contacts...)
- Utilisé surtout pour "peupler" certains composants XUL : liste, tree, menu etc..
- Comment : attribut "datasource" + balises <template> pour spécifier un modèle XUL



- Javascript 1.5, implémentation de l'EcmaScript, normalisé par l'ECMA
- Langage de script orienté objet
- De nombreux objets sont fournis de base :
  - objets DOM (Element, Node, TreeWalker..)
  - web service (SOAP, WSDL)
  - xml (xmlHttpRequest, XMLDocument, XPathEvaluator, XSLTProcessor ...)
  - objets représentant des composants HTML, XUL et styles CSS
  - plus les composants XPCOM

## Utiliser les technologies de Mozilla pour des applications Web et desktop



# Applications Web vs Sites Web

## Sites Web :

- But : publier de l'information
- Formulaire pour envoyer des informations (forum, demande de renseignements...)
- Technologie : HTML

## Applications Web :

- Gérer des données, des informations
  - Remplace les applications client (lourd) – serveur traditionnel
  - ex : gérer des stocks, des produits, des informations etc..
  - phpGroupWare, phpMyAdmin, Admin SPIP etc...
- Technologie : HTML à défaut d'autre chose, **mais peu adapté pour réaliser une interface utilisateur ergonomique et efficace**





# Applications Web traditionnelles

Actuellement :

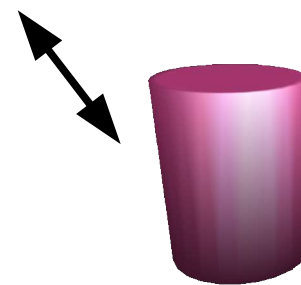
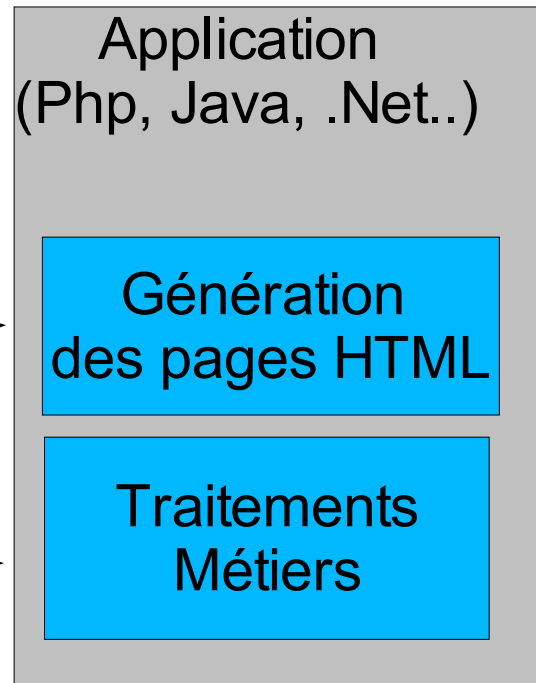


HTML, CSS, JS

Contenus formulaires,  
requête affichage de page



Serveur  
http



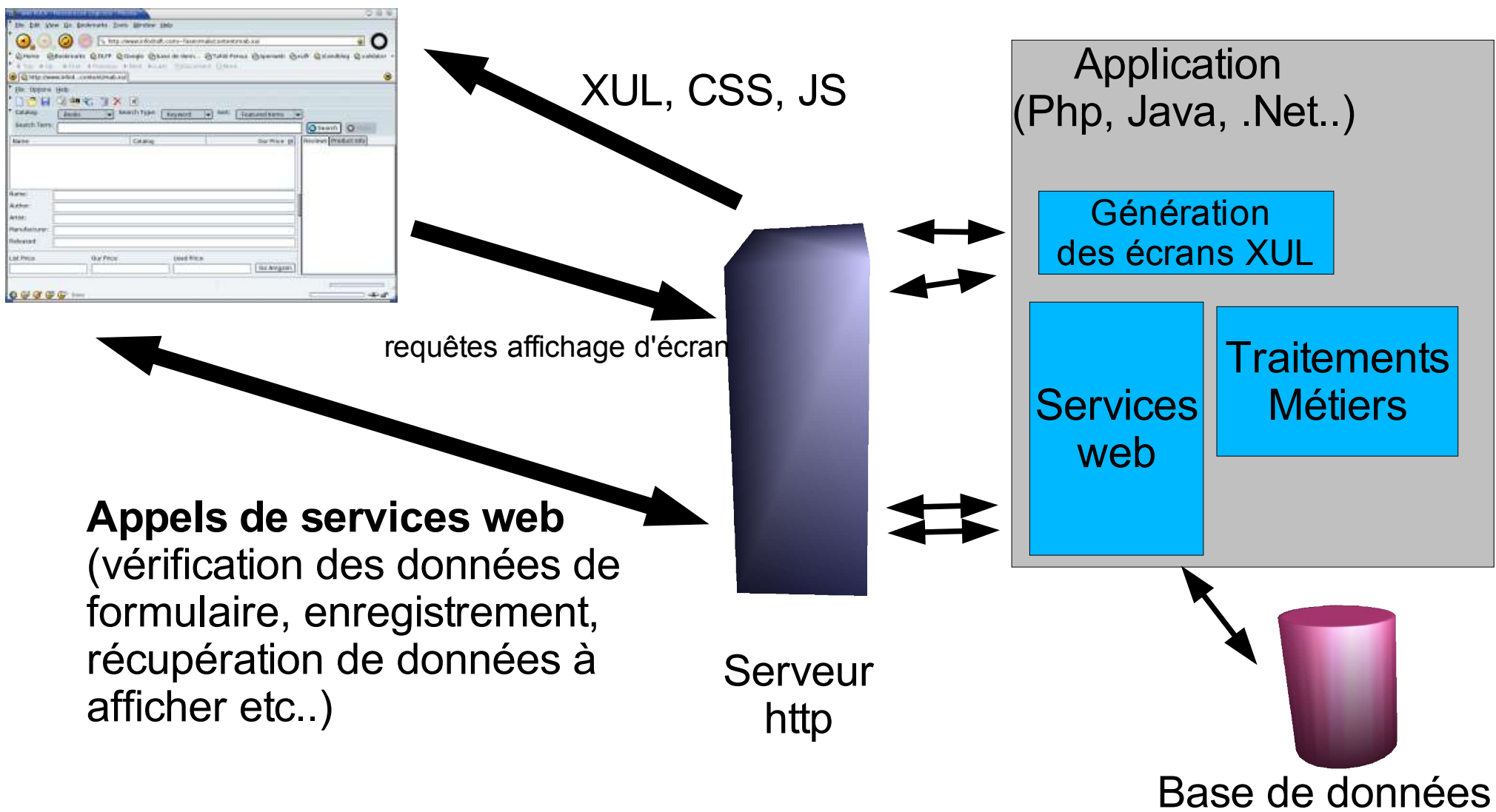
Base de données





# Applications web avec XUL

Remplaçons HTML par XUL :



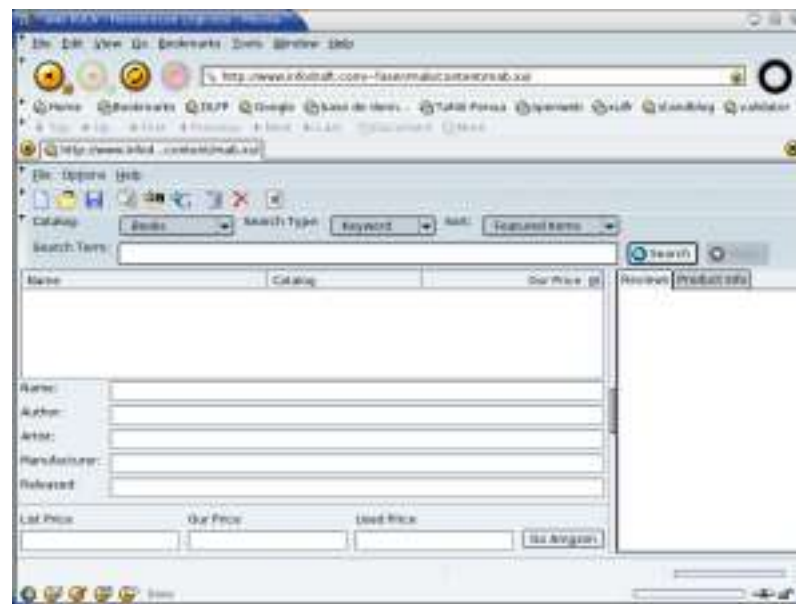


# Applications Web avec XUL

- Ergonomie équivalente à de vrais applications "desktop"
- Cela reste du client léger mais "riche"
- -> Rien à installer sur le poste client (sauf Mozilla ;-)
- Look personnalisable grâce à CSS
- Utilisation des web services : pas besoin de régénérer l'interface à chaque action = Déplacement de la logique applicative coté client sans pour autant l'alourdir -> serveur web moins sollicité.
- Séparation plus nette entre interface (son comportement) et les traitements métiers



- Mozilla Amazon Browser <http://mab.mozdev.org>



- Il est possible de
  - signer l'application pour plus de sécurité (certificats)
  - l'installer en locale
- Avantages :
  - sécurité
  - privilèges sur le poste utilisateur : accès aux objets XPCOM
- utilisation du format XPI = format d'archive de Mozilla
  - archive zip = fichiers applis + fichiers manifest en rdf
  - un lien dans une page vers un xpi = installation en 1 clic



- <http://www.mozilla.org> : site de la fondation
- <http://www.mozilla-europe.org>
- <http://www.xulplanet.com> : **LA** référence pour les développeurs (en anglais)
- <http://www.xulfr.org> : site francophone sur le sujet.
- <http://www.mozdev.org> : applications et extensions mozilla
- [http://xulfr.org/docs/conf\\_libreast\\_2004/](http://xulfr.org/docs/conf_libreast_2004/) : cette présentation



FIN

