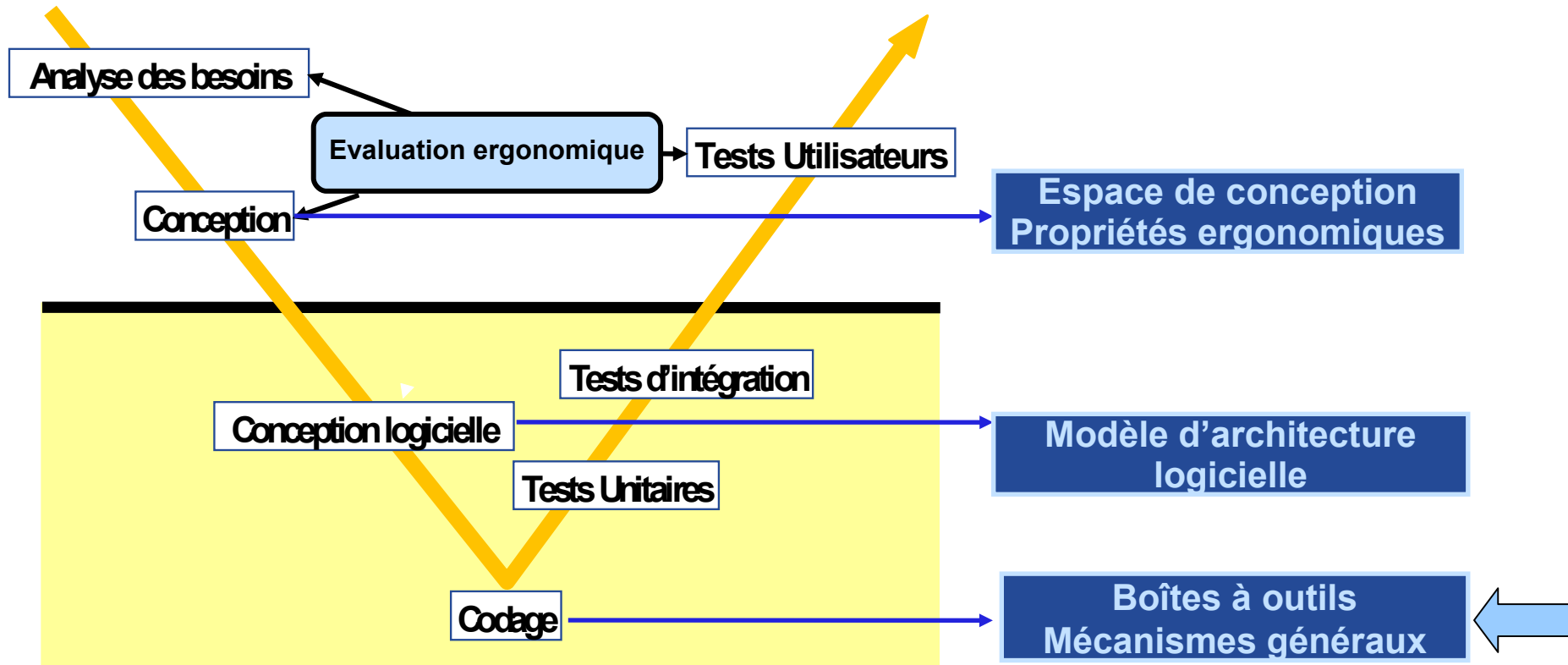


Web 2.0 : Impacts sur les IHMs

Module Plasticité
SI5 - Master IFI

Plan du cours

- Nouveaux usages d'Internet => naissance du Web2.0
- Impacts sur les IHM => naissance des interfaces dites "riches", les RIA (Rich Internet Applications)
- Vers un Web3.0 ?



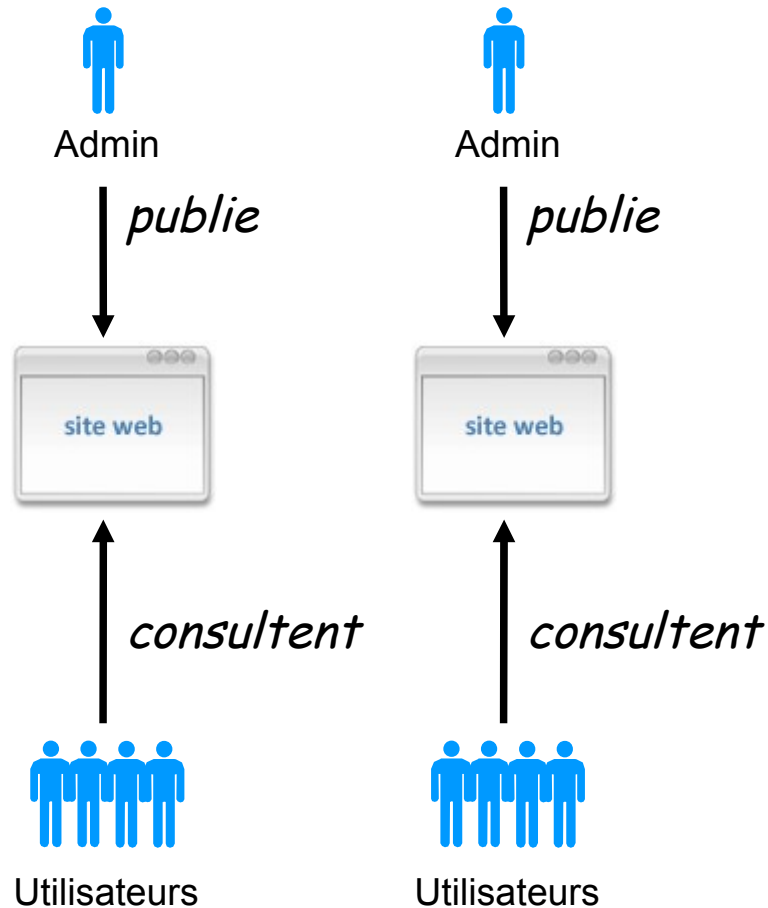
IHM du web2.0 & plasticité

- Plasticité ?
 - capacité d'adaptation d'une IHM : à la conception / exécution
 - au contexte d'usage : langage, plate-forme logicielle/matérielle, utilisateur, environnement, ...
 - dans le respect de la valeur attendue par l'utilisateur cible : utilisabilité, ergonomie
- Quelles formes de plasticité pour les IHM du web2.0 ?

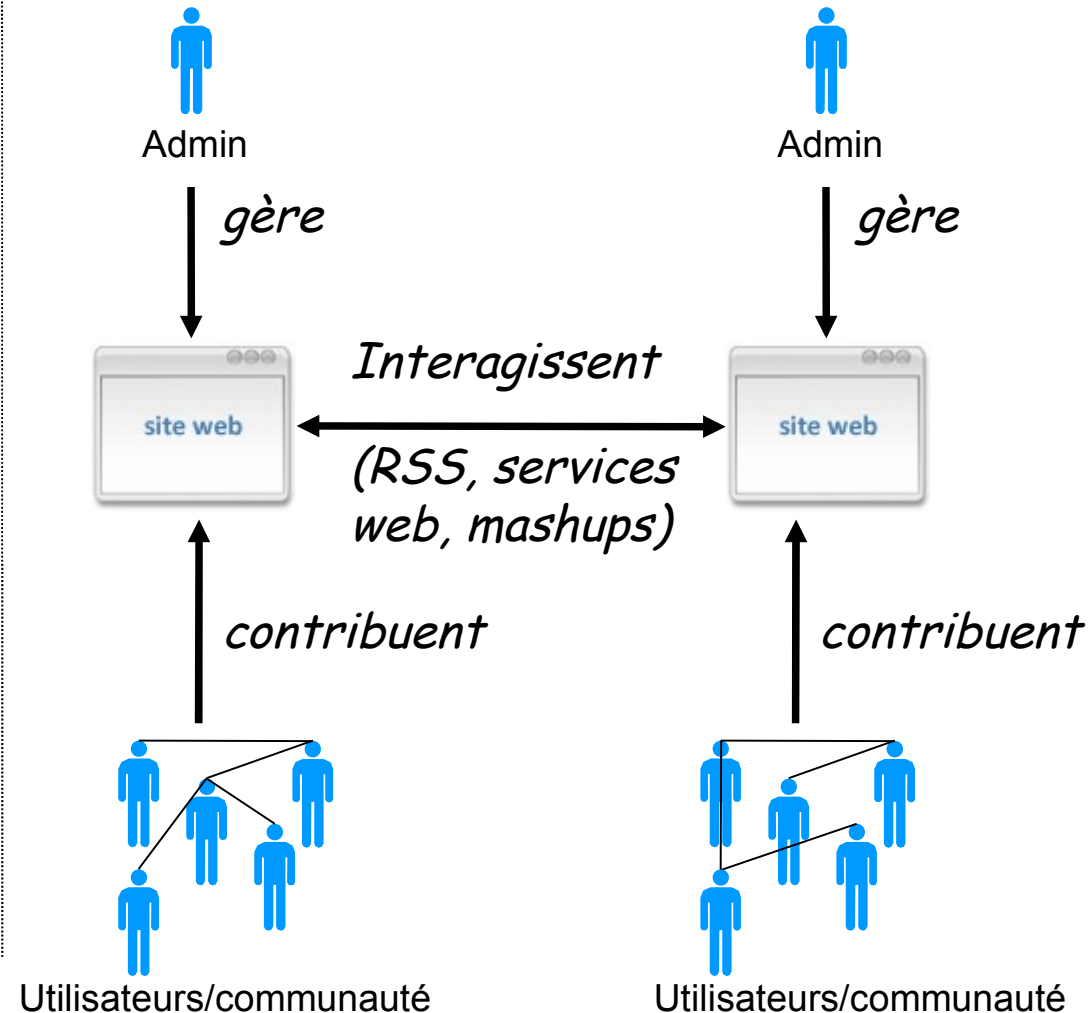
Web2.0 : nouveaux usages

Evolution d'Internet : point de vue des usages

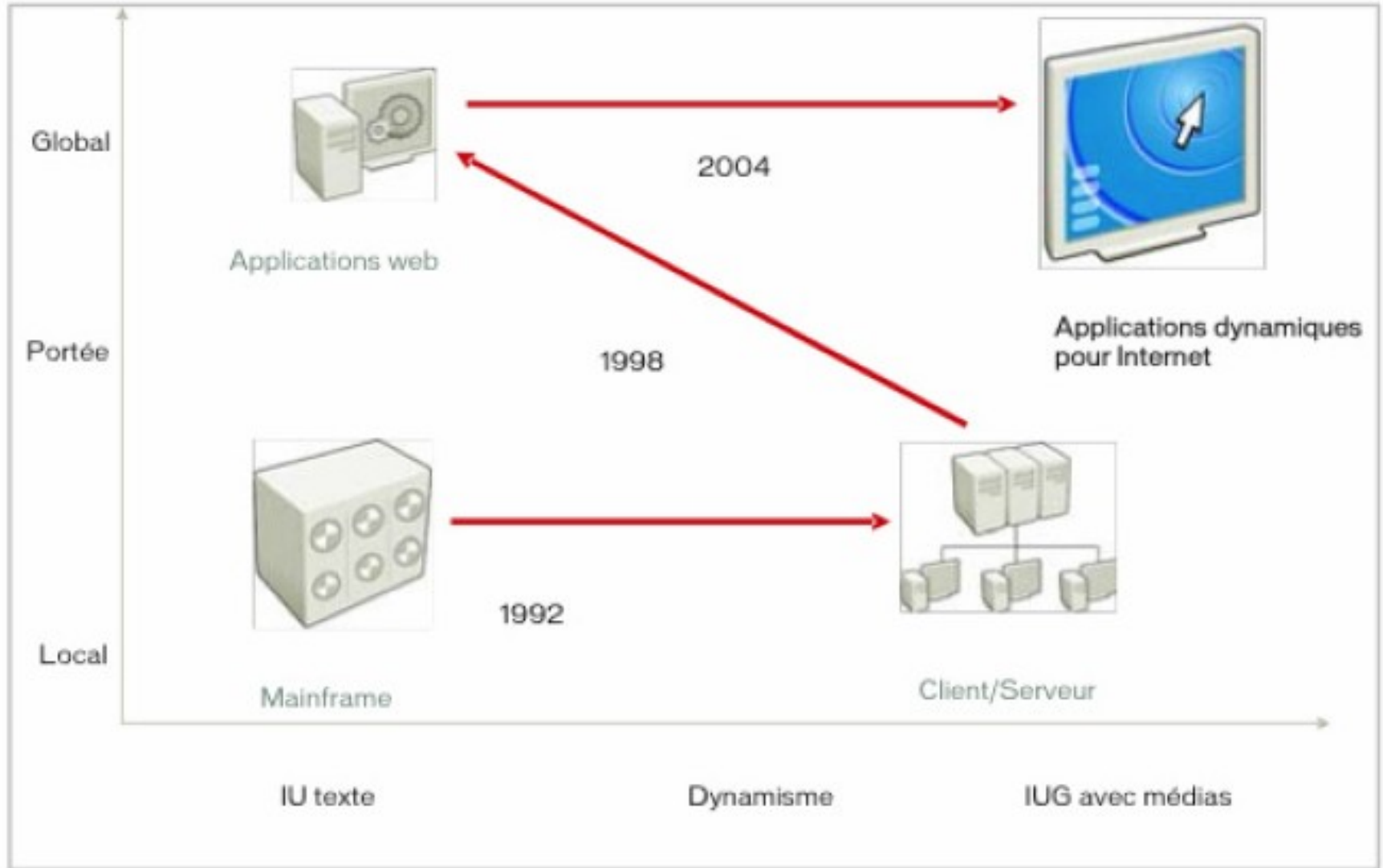
Web1.0



Web2.0



Évolution des IHMs



source: adobe.org

Web2.0, définition et points clé

- Met l'accent sur l'agilité et la réutilisabilité des applis :
 - Focalisation sur les services rendus et sur les données mais pas sur les applications
 - Faiblement couplé pour faciliter l'assemblage de services en applications
 - Faciliter la réutilisabilité des contenus
- Implication des utilisateurs comme acteurs du service rendu
 - Plus le service est utilisé, plus il s'améliore
 - Plus le service est utilisé, plus il devient intéressant
- **Le Web2.0 qualifie les interfaces permettant aux internautes d'interagir sur le contenu des pages et entre eux**

Du web1.0 vers le web2.0 : synthèse

Depuis...

- Contemplatif
- Contenu propriétaire
- Tri arborescent (taxonomie)
- "Webscrapping"
- Sites perso

- Britannica online
- Ofoto
- mp3.com
- amazon
- Google



...Vers...

- Participatif
- Contenu collectif (wikis)
- Tri par tags ("folksonomy")
- Services web & mashups
- Blogs, réseaux sociaux

- Wikipedia
- Flickr
- Napster
- eBay
- iGoogle



Enrichissement des interfaces (RIA)

Les interfaces riches

Jeu des 7 différences ...

Informations sur la chambre

Quand souhaiteriez-vous séjourner au Walt Disney World Swan ? Saisissez votre date d'arrivée prévue et le nombre de nuits que vous souhaitez passer à l'hôtel, ou saisissez les dates d'arrivée et de départ.

Arrivée

et

Durée du séjour (nuits)

ou

Départ

Préférences

Chambre(s)

Nombre d'adultes par chambre

Informations sur les tarifs (facultatif)

Notre hôtel est entièrement non-fumeur.

☐ Accès pour chaises roulantes requis

Type de lit

[Nouvelle Recherche](#)

[Continuer](#)

Internet

Internet

100%

Swissotel Chicago - Windows Internet Explorer

https://reservations.hotelier.com/istay.cfm?hotelid=14124

Swissotel Chicago

swissotel CHICAGO

Home Travel Agent Group Reservations Corporate Rates

Select Check-in and Check-out Dates

April 2009 - May 2009

April 2009

May 2009

Restrictions apply, mouseover for details.

Check-in: May 6, 2009 Adults: 1

Check-out: May 8, 2009 Children: 0-17 yrs 0

Nights: 2

Room Type: Navy Pier Lakeview

Select Just a Room

Select a Room Best Available Rate

Superior King USD249.00

Navy Pier Lakeview Room USD299.00

Corner King Deluxe USD299.00

Executive Suite Not available

Enhance Your Stay:

Rollaway Bed USD100.00

Valet Parking USD98.00

Enhancement Subtotal: USD0.00

USD Subtotal: USD598.00

Continue This Reservation

SWISSOTEL CHICAGO | 323 E. WACKER DRIVE | CHICAGO | IL | 60601 | RESERVATIONS: 888-73-SWISS | WWW.SWISSOTELCHICAGO.COM



Spectacular views of Lake Michigan, the Chicago River and Navy Pier. Oversized room with marble bath, separate shower, work area, coffee maker, minibar, hair dryer, and plush robe.

Terms & Conditions

Terminé

Définitions

- A quoi ressemble l'IHM d'une application web traditionnelle ?
 - Une page web de type formulaire (HTML, ASP, JSP, PHP, etc)
 - Le navigateur n'est qu'un "lecteur" universel de pages web

The screenshot shows a web form for hotel booking. It is divided into several sections: 'Informations sur la chambre', 'Préférences', and 'Informations sur les tarifs (facultatif)'. The first section asks for arrival and departure dates, with dropdown menus for month, day, and year. The second section asks for the number of rooms and adults per room. The third section includes a checkbox for wheelchair access and a dropdown for bed type. At the bottom, there are buttons for 'Nouvelle Recherche' and 'Continuer'.

Informations sur la chambre

Quand souhaiteriez-vous séjourner au Walt Disney World Swan ? Saisissez votre date d'arrivée prévue et le nombre de nuits que vous souhaitez passer à l'hôtel, ou saisissez les dates d'arrivée et de départ.

Arrivée

et

Durée du séjour (nuits) 

ou

Départ

Préférences

Chambre(s)

Nombre d'adultes par chambre

Informations sur les tarifs (facultatif)

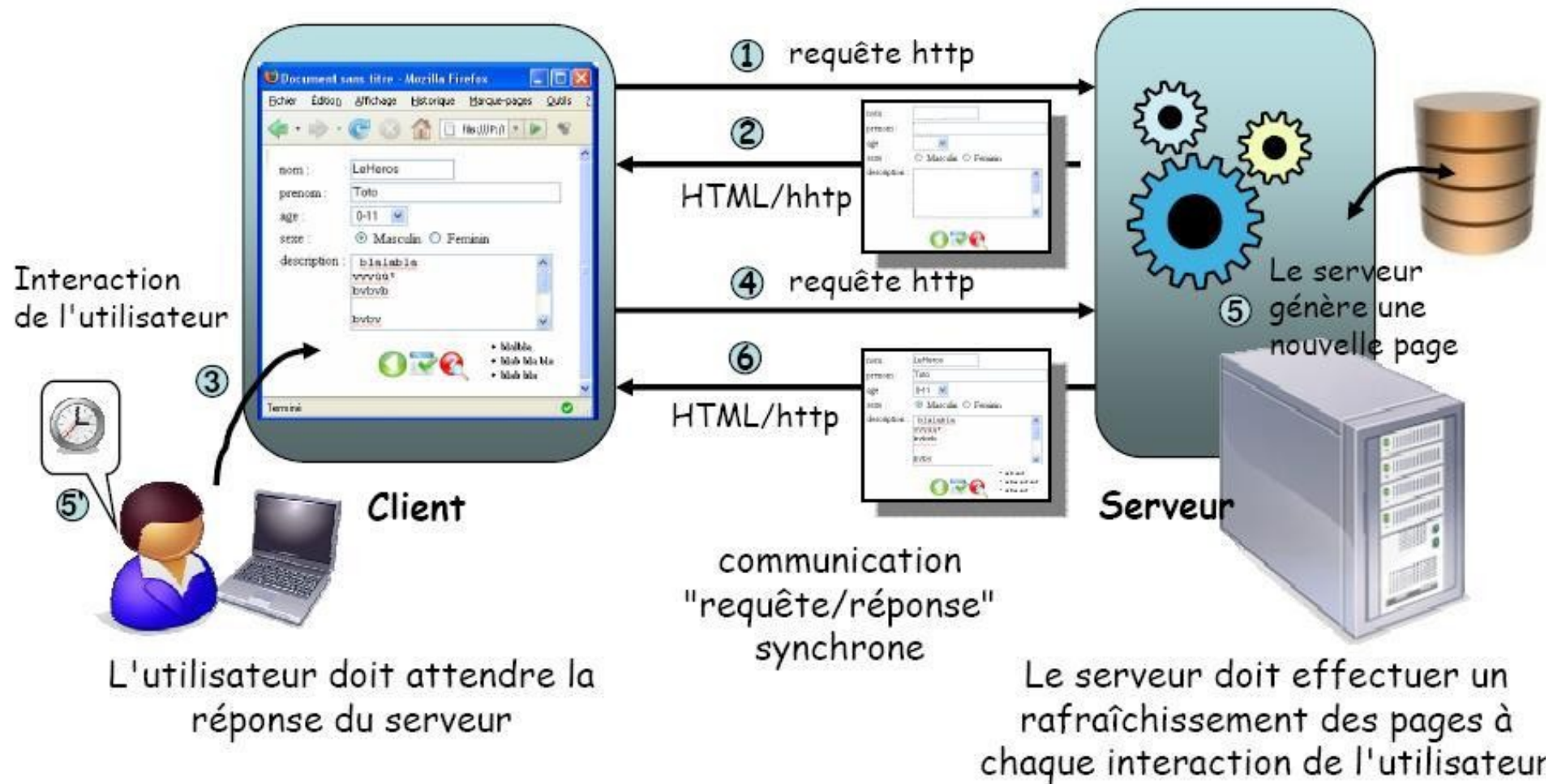
Notre hôtel est entièrement non-fumeur.

☐ Accès pour chaises roulantes requis

Type de lit

Nouvelle Recherche **Continuer**

Interactions via une application web "traditionnelle"



source : extrait du cours de Philippe Genoud

Limites ergonomiques du web1.0

- Constats :
 - Enchaînement de pages décidé par le serveur (aucun traitement coté client)
 - Nouveau contenu = rafraichissement de la page
 - Liste de widgets et navigation limitées (click/tab)
- Conséquence N°1 : Navigation à travers une série de pages pour remplir une simple tâche et interactions non continues
- Conséquence N°2 : Inactivité forcée de l'utilisateur durant le traitement d'une requête
- Conséquence N°3 : Exploration interactive des données impossible (pas de retour d'information immédiat)
- Conséquence N°4 : Perte du contexte lié au traitement d'une tache suite au rafraîchissement

Jeu des 7 différences ...

Informations sur la chambre

Quand souhaiteriez-vous séjourner au Walt Disney World Swan ? Saisissez votre date d'arrivée prévue et le nombre de nuits que vous souhaitez passer à l'hôtel, ou saisissez les dates d'arrivée et de départ.

Arrivée

et

Durée du séjour (nuits)

ou

Départ

Préférences

Chambre(s)

Nombre d'adultes par chambre

Informations sur les tarifs (facultatif)

Notre hôtel est entièrement non-fumeur.

☐ Accès pour chaises roulantes requis

Type de lit

[Nouvelle Recherche](#)

[Continuer](#)

Internet

Swissotel Chicago - Windows Internet Explorer

https://reservations.hotelier.com/istay.cfm?hotelid=14124

Swissotel Chicago

swissotel CHICAGO
where the night really has taste™

Home Travel Agent Group Reservations Corporate Rates

Select Check-in and Check-out Dates

April 2009 - May 2009

Sun Mon Tue Wed Thu Fri Sat

April 2009

May 2009

Check-in: May 6, 2009 Adults: 1
Check-out: May 8, 2009 Children: 0-17 yrs 0
Nights: 2
Room Type: Navy Pier Lakeview

Select Just a Room

Select a Room Best Available Rate

Average Daily Rate

Superior King USD249.00

Navy Pier Lakeview Room USD299.00

Corner King Deluxe USD299.00

Executive Suite Not available

Enhance Your Stay:

☐ Rollaway Bed USD100.00

☐ Valet Parking USD98.00
Unlimited in and out privileges

Enhancement Subtotal: USD0.00

USD Subtotal: USD598.00

[Continue This Reservation](#)

SWISSOTEL CHICAGO | 323 E. WACKER DRIVE | CHICAGO | IL | 60601 | RESERVATIONS: 888-73-SWISS | WWW.SWISSOTELCHICAGO.COM



Spectacular views of Lake Michigan, the Chicago River and Navy Pier. Oversized room with marble bath, separate shower, work area, coffee maker, minibar, hair dryer, and plush robe.

[Terms & Conditions](#)

Réservation d'hotel & expérience utilisateur

- **Version HTML**

- Plusieurs pages pour effectuer la réservation de bout en bout



- **Version Flash**

- Un seul écran interactif



- **TravelClick/Webvertising iHotelier**

- "75% of users choose OneScreen in favor of the HTML version"
- Reduces the time it takes to make a reservation from at least three minutes to under one

Source: <https://reservations.ihotelier.com/istay.cfm?hotelid=14124>

http://www.adobe.com/cfusion/showcase/index.cfm?event=casestudydetail&casestudyid=2486&loc=en_us

Mariage de 2 types d'applications opposées

Applications sur "bureau" (standalone/desktop)

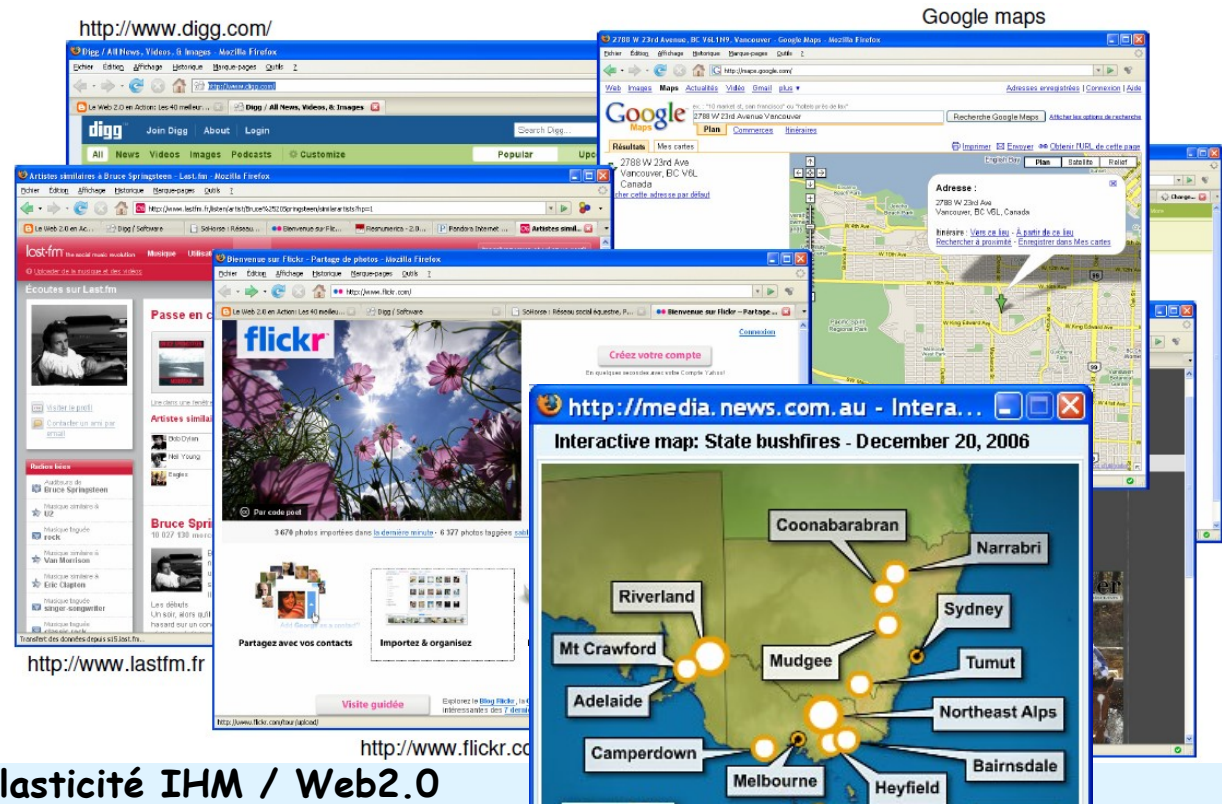
- Déconnectées
- Offrant de multiples possibilités en terme d'affichage/rendu/interaction
- Exécutable à installer sur l'ordinateur
- Une version par OS
- Accès libre aux ressources de l'ordinateur
- Assurant la confidentialité des données

Applications web

- Fortement connectées (communication avec un serveur distant)
- Rendu et interactions limités
- Pas d'installation/mise à jour
- Portables
- Cookies seulement
- Données susceptible d'être interceptées

Définitions

- A quoi ressemble l'IHM d'une application web traditionnelle ?
 - Une page web de type formulaire (HTML, ASP, JSP, PHP, etc)
- Qu'est-ce qu'une RIA (Rich Internet application) ?
 - Terme introduit par Macromédia en 2004
 - Une application dont l'IHM se rapproche d'une application "desktop" classique mais qui s'exécute dans un simple navigateur web



Idées derrière les interfaces riches

- La logique de présentation est déplacée côté client
- L'état du service (les données) reste côté serveur
 - Toutes les interactions de l'utilisateur ne nécessitent pas l'interrogation du serveur (échanges client/serveur limités)
 - L'utilisateur continue ses activités durant le traitement d'une requête (communication asynchrone)
 - Après le chargement initial de la page, échange de données (i.e. : sans redessiner entièrement la page)

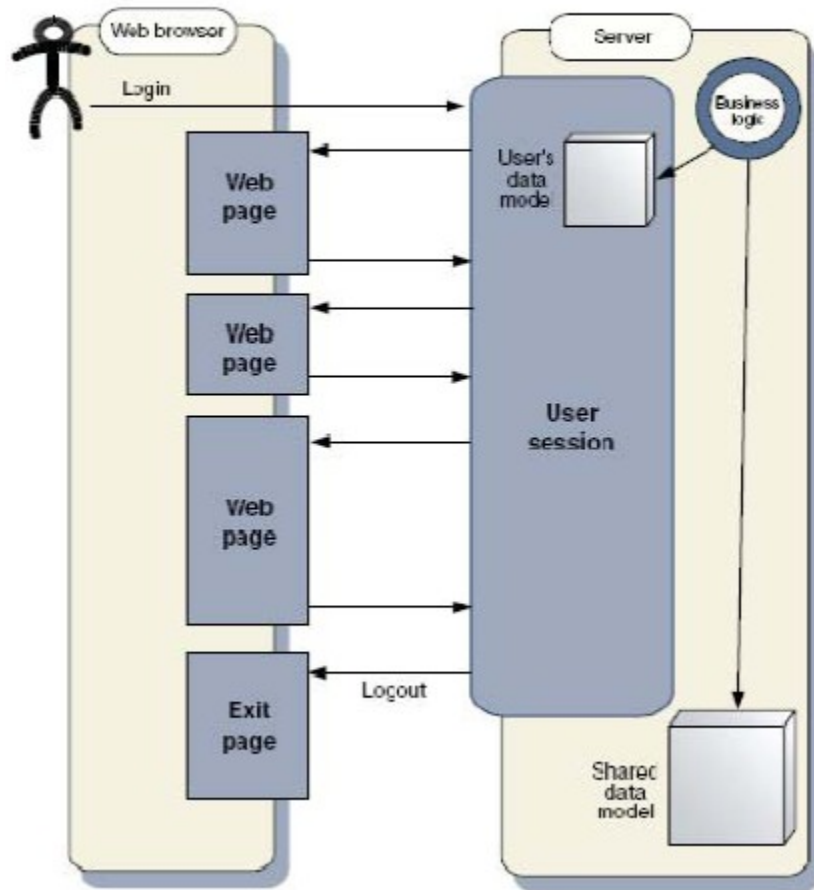


Gains en ergonomie, interactivité, rapidité, bande passante, charge serveur, etc

Interactions avec le serveur web

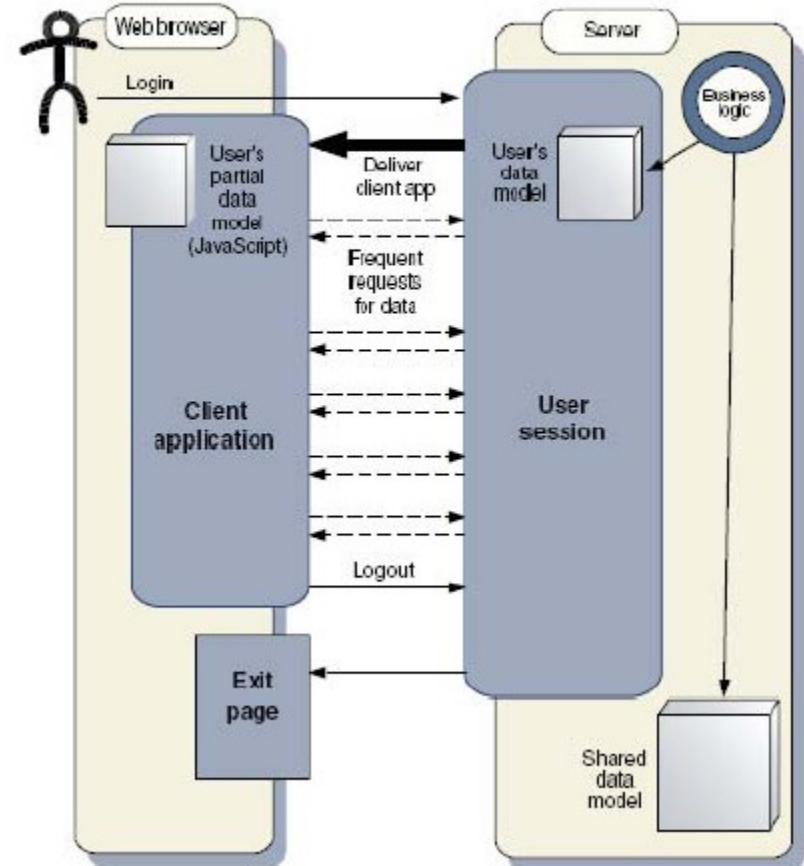
Application web traditionnelle

Infrequent large updates



Application web riches

Frequent small updates

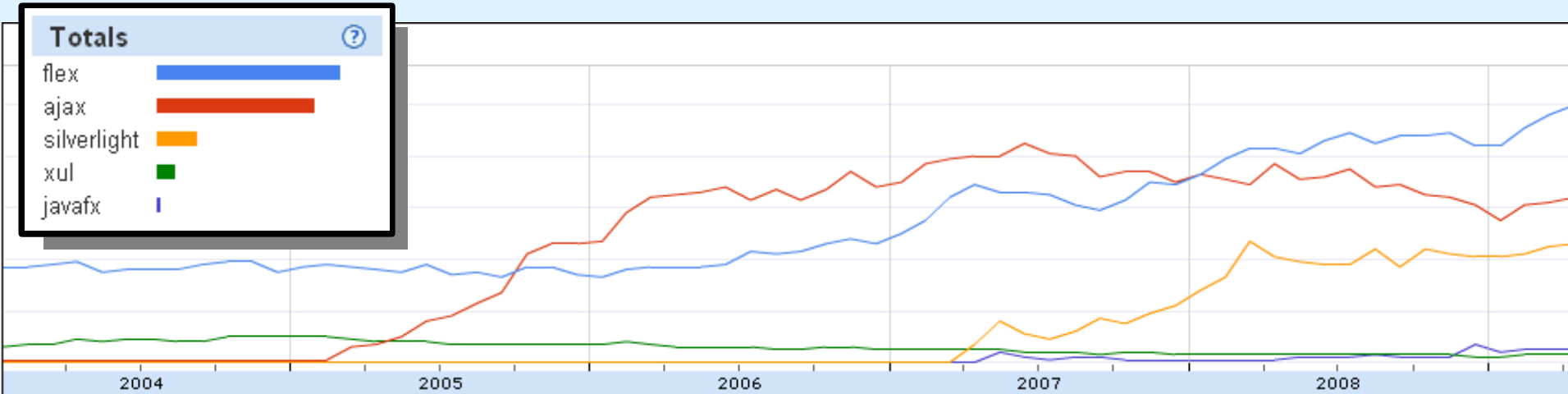


source: "Ajax in Action" by Dave Crane et al, Manning Press

Solutions RIAs disponibles

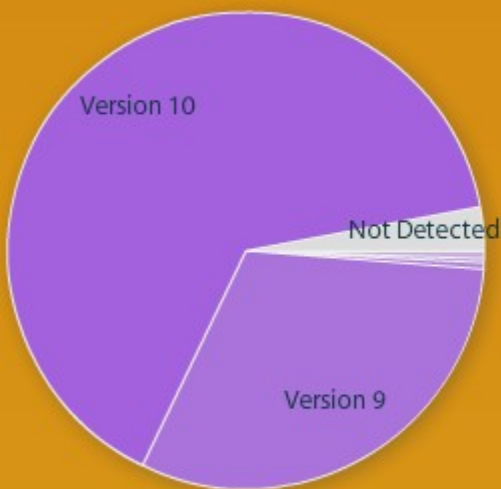
- **AJAX** : un ensemble de techno open source éprouvées
 - Asynchronous Javascript And XML
 - Utilisation combinée nouvelle
- **Autres offres**
 - Adobe Flex (2004) :
<http://www.adobe.com/support/documentation/en/flex/>
 - Microsoft Silverlight (2006) :
<http://www.silverlight.net>
 - Sun JavaFX (2008) :
<http://www.javafx.com/>
 - Mozilla XUL (XML User Interface Language)
<http://www.mozilla.org/projects/xul/>
 - ...

La tendance actuelle ...

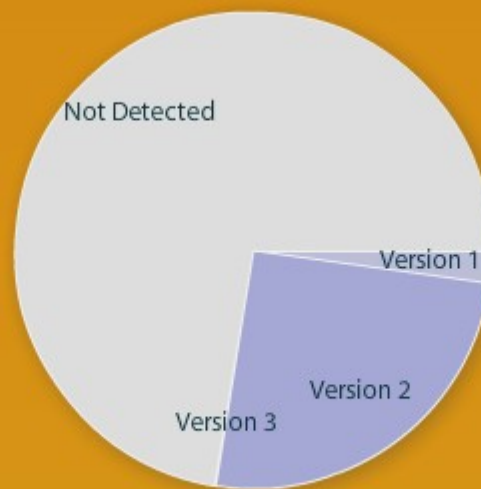


Source : Google Insights

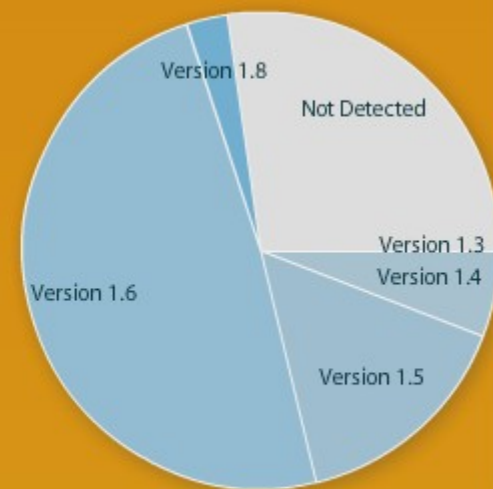
Adobe Flash/Flex Player



Microsoft Silverlight Player



Sun Java

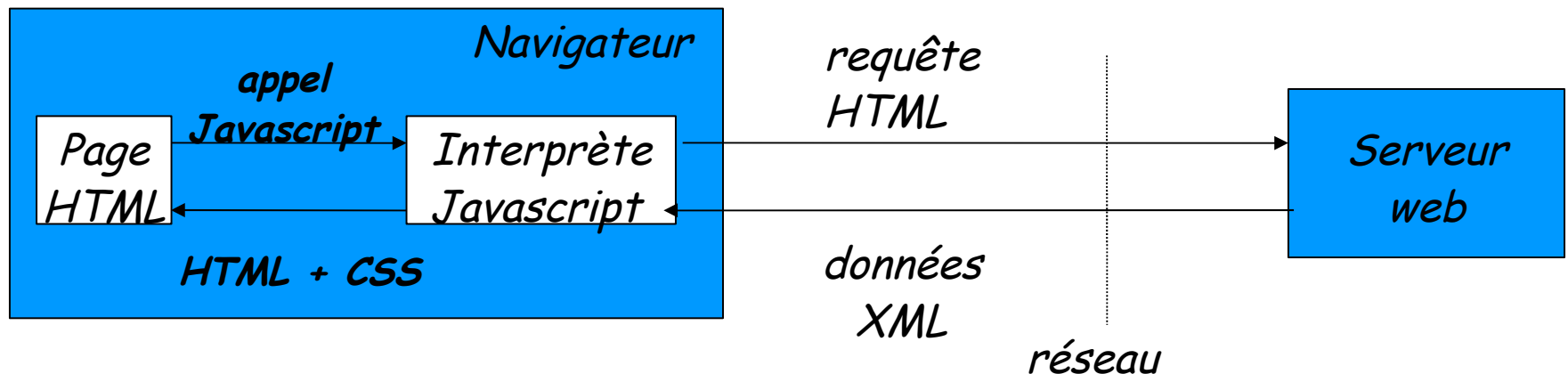


Statistics collected from 1,623,834 daily unique browsers across 55 sites over the past 30 days.

Source : <http://www.riastats.com/>

AJAX en bref

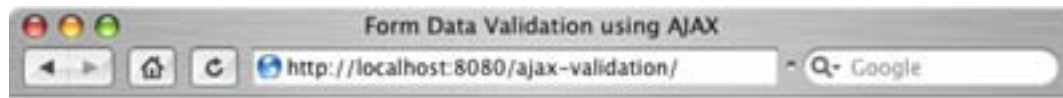
- 1er échange : le client accède à une URL et le serveur envoie une page + du code javascript
- Échanges suivants : Le client peut traiter des actions localement grâce au javascript ou bien lancer une requête HTTP asynchrone lorsque l'intervention du serveur est nécessaire
- Mise à jour partielle de l'interface coté client avec des données du serveur au format XML ou (texte)



AJAX = XMLHttpRequest + DHTML

- DHTML = Pages web dynamiques grace à l'utilisation conjointe de Javascript, DOM et CSS
- **Javascript**
 - intervient lorsqu'un événement est déclenché sur la page
 - sert de "glue" entre les différentes briques
- **DOM (Document Object Model)**
 - structure les pages web sous forme arbres
 - permet d'accéder/mettre à jour le contenu/structure/style des pages
- **CSS (Cascading Style Sheets)**
 - permet une séparation contenu (types des éléments)/forme (apparence des éléments) de l'IHM
 - modifiable par le code Javascript via DOM
- **XMLHttpRequest** pour les communications asynchrone avec le serveur (utilisé dans le code Javascript)

Exemple : Validation coté serveur sans rechargement de la page

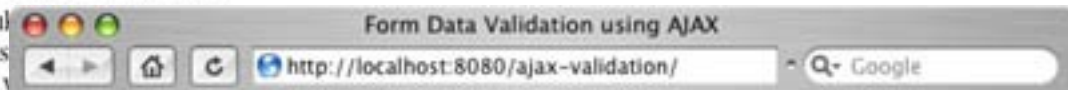


Form Data Validation using AJAX

This example shows how you can use AJAX to do server-side form data validation without a page reload.

In the form below enter a user id. By default the user ids "greg" and "duke" are taken. If you attempt to enter a user id that has been taken an error message will be displayed next to the form field and the "Create Account" button will be disabled. After entering a valid user id and selecting the "Create Account" button that user id will be added to the list of user ids that are taken.

User Id: Valid User Id



Form Data Validation using AJAX

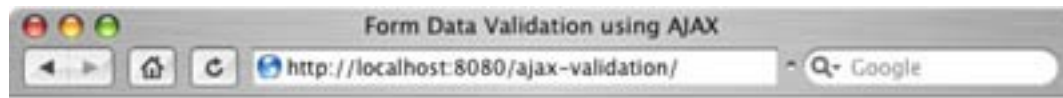
This example shows how you can use AJAX to do server-side form data validation without a page reload.

In the form below enter a user id. By default the user ids "greg" and "duke" are taken. If you attempt to enter a user id that has been taken an error message will be displayed next to the form field and the "Create Account" button will be disabled. After entering a valid user id and selecting the "Create Account" button that user id will be added to the list of user ids that are taken.

User Id: Invalid User Id

Source : <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/RealtimeValidation/>

Exemple : Validation coté serveur sans rechargement de la page

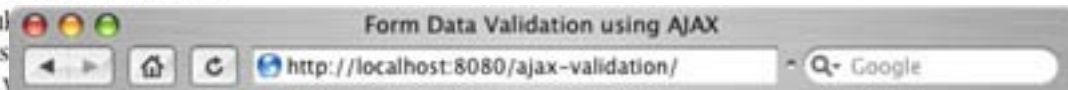
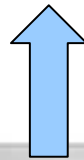


Form Data Validation using AJAX

This example shows how you can use AJAX to do server-side form data validation without a page reload.

In the form below enter a user id. By default the user ids "greg" and "duke" are taken. If you attempt to enter a user id that has been taken an error message will be displayed next to the form field and the "Create Account" button will be disabled. After entering a valid user id and selecting the "Create Account" button that user id will be added to the list of user ids that are taken.

User Id: Valid User Id

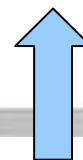


Form Data Validation using AJAX

This example shows how you can use AJAX to do server-side form data validation without a page reload.

In the form below enter a user id. By default the user ids "greg" and "duke" are taken. If you attempt to enter a user id that has been taken an error message will be displayed next to the form field and the "Create Account" button will be disabled. After entering a valid user id and selecting the "Create Account" button that user id will be added to the list of user ids that are taken.

User Id: Invalid User Id



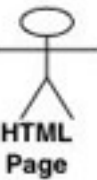
Source : <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/RealtimeValidation/>

Interactions et propagation des événements

L'utilisateur tape un nom



L'action est transformée en événement



HTML Page

onkeyup event

Préparation requête

JavaScript

<<create>>

URL =validate?id=duke

set callback function

XMLHttpRequest

Envoi requête

Get validate?id=duke

Coté serveur

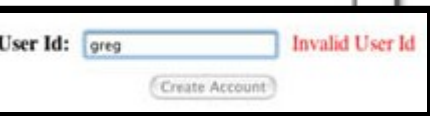
ValidationServlet
<<mapped to validate>>

call callback function

Rappel asynchrone du client

update DOM

Mise à jour de la page



Source : <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/RealtimeValidation/>

Manipulation de la page web avec DOM

...

```
<b>User Id:</b>
```

```
<FORM method="post" action="">
```

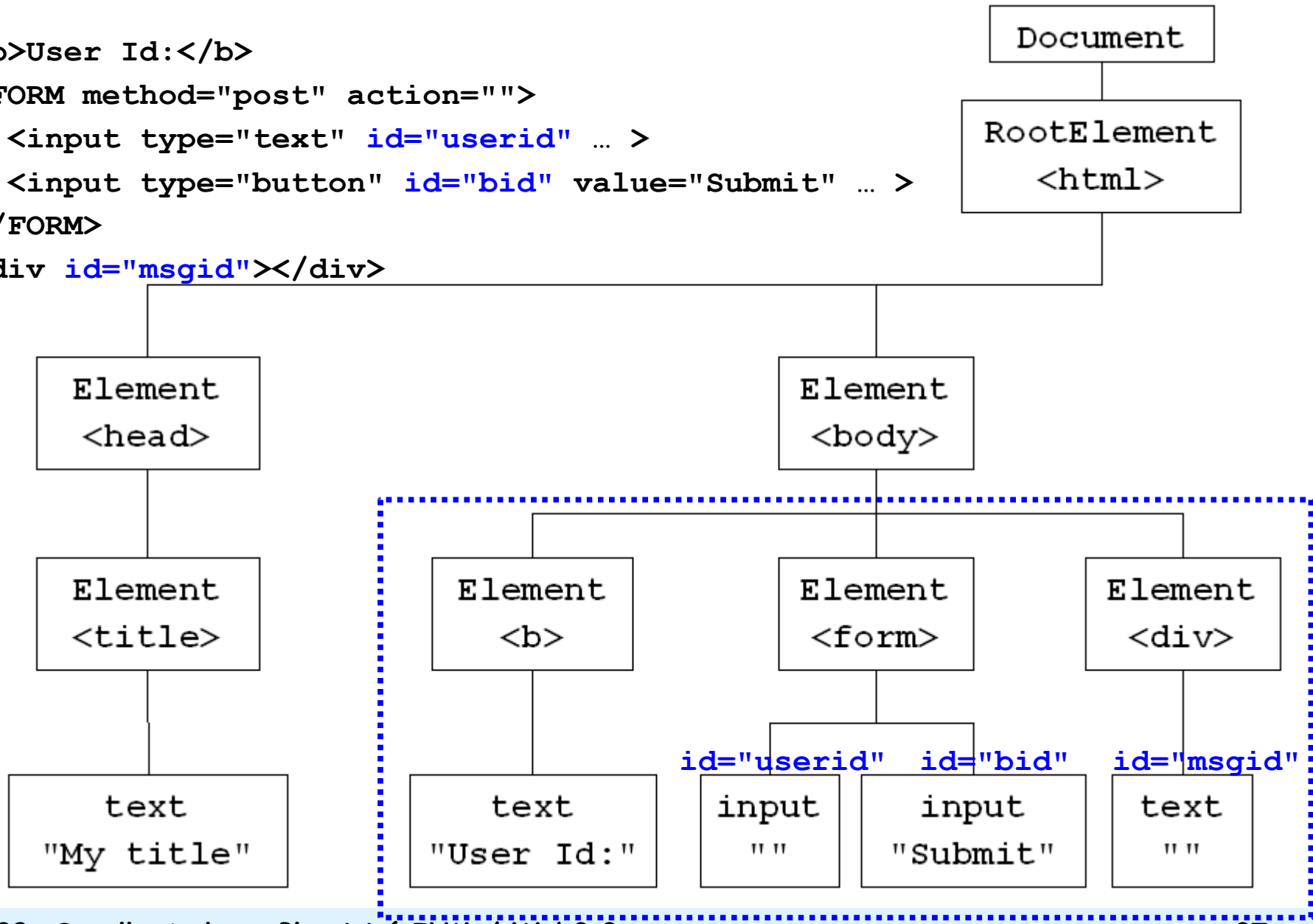
```
  <input type="text" id="userid" ... >
```

```
  <input type="button" id="bid" value="Submit" ... >
```

```
</FORM>
```

```
<div id="msgid"></div>
```

...



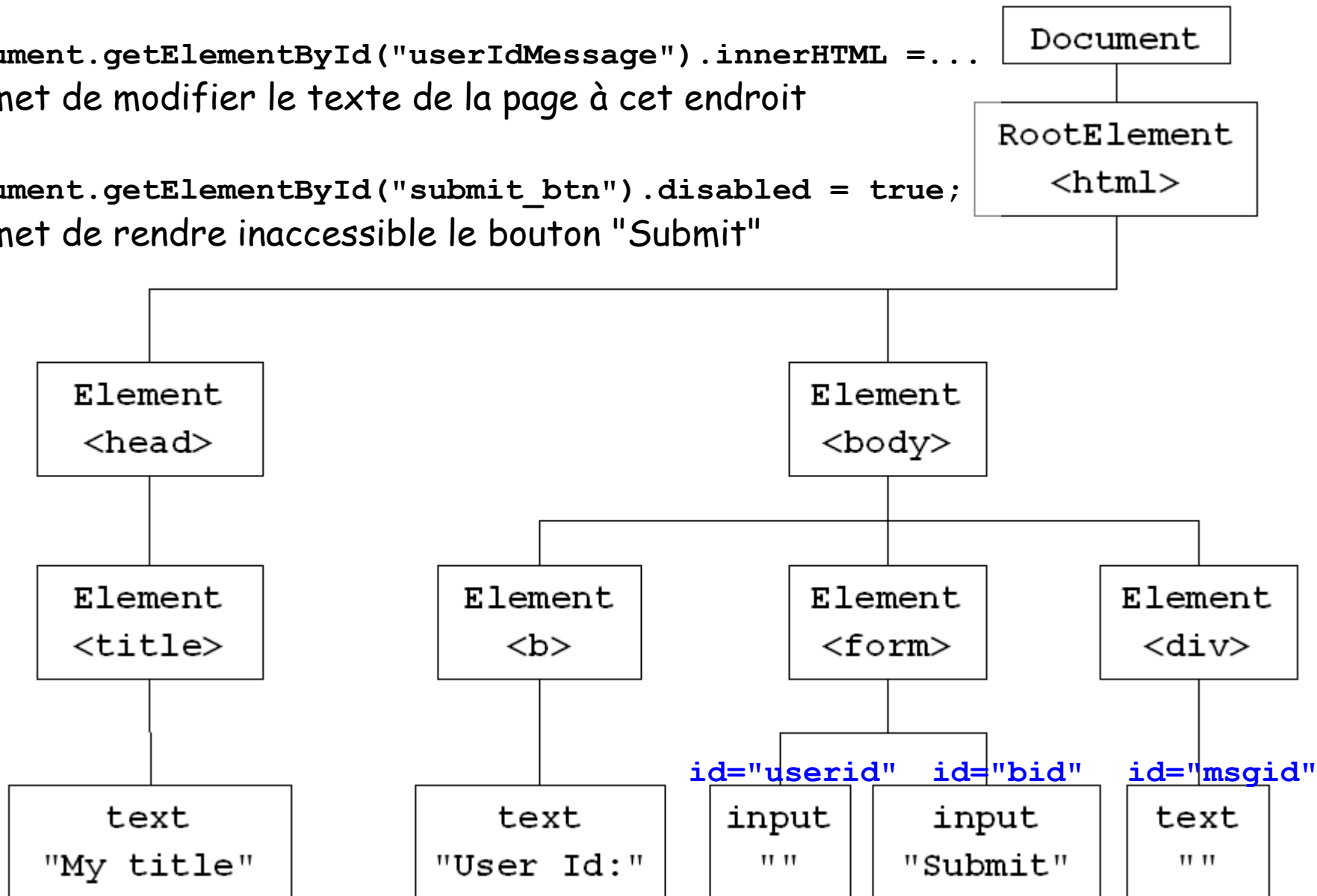
Manipulation de la page web avec DOM

```
document.getElementById("userIdMessage").innerHTML = ...
```

Permet de modifier le texte de la page à cet endroit

```
document.getElementById("submit_btn").disabled = true;
```

Permet de rendre inaccessible le bouton "Submit"



Étapes de développement typiques avec Ajax

1. Association d'un événement à une action utilisateur au travers d'une widget
2. Implémentation d'une fonction javascript associée au déclenchement de l'événement
 - (a) Création et configuration d'un objet XMLHttpRequest
 - (b) Envoi d'une requête asynchrone via l' objet XMLHttpRequest
3. Implémentation d'une fonction de callback (utilisée au moment de la configuration lors de l'étape 2) appelée lorsque le serveur renvoie sa réponse
 - (a) Traitement de la réponse du serveur
 - (b) Mise à jour de la page HTML (DOM)
- *. Traitement coté serveur + résultat sous forme d'un document XML (ou texte)

Hello world

■ Scenario :

1. L'utilisateur tape une chaine de caractères et valide en appuyant sur le bouton
2. Le message "Bonjour" concaténé à la chaine de caractères qu'il vient de taper est affiché sur la page

 ... ligne de texte qui sera rafraichie ...	tout le monde ... ligne de texte qui sera rafraichie ...	tout le monde Bonjour tout le monde
Submit	Submit	Submit

■ Traitement coté serveur :

Lecture d'un fichier et renvoie du résultat ("Bonjour")

■ Traitement coté client :

Concaténation du résultat à la chaine de caractères et mise à jour de la page

➔ Plein de variantes possibles !

Hello World en AJAX "ad-hoc"

```
<html>
  <head>
    <script language="JavaScript">
      function submitForm() {
        ... // voir page suivante
      }
    </script>
    <!-- Personnalisation des widgets avec CSS -->
    <style type="text/css">
      @import "monstyle.css";
    </style>
  </head>

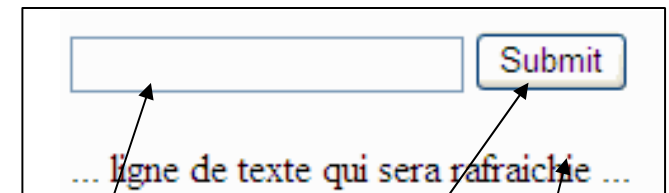
  <body>
    <FORM name="ajax" method="post" action="">
      <input type="text" name="texte" size="32" value="">
      <input type="button" value="Submit" onclick="submitForm()">
    </FORM>
    <div id="zone"> ... ligne de texte qui sera rafraichie ... </div>
  </body>
</html>
```

```
body { // fichier css
  background-color: white;
  color: blue;
  font-family: Comic Sans MS;
  font-size: small;
}
input[type=button] {
  color:#050;
  font: bold 84% Comic Sans MS;
  background-color:#fed;
  border: 1px solid;
  border-color: #696 #363 #363 #696;
  filter:progid:DXImageTransform.Microsoft.Gradient
  (GradientType=0,StartColorStr='#ffffff',EndColorStr='#ffa0a0');
}
```

Hello World en AJAX "ad-hoc"

```
<html>
<head>
  <script language="JavaScript">
    function submitForm() {
      ... // voir page suivante
    }
  </script>
  <!-- Personnalisation des widgets avec CSS -->
  <style type="text/css">
    @import "monstyle.css";
  </style>
</head>

<body>
  <FORM name="ajax" method="post" action="">
    <input type="text" name="texte" size="32" value="">
    <input type="button" value="Submit" onclick="submitForm()">
  </FORM>
  <div id="zone"> ... ligne de texte qui sera rafraichie ... </div>
</body>
</html>
```



Hello World en AJAX "ad-hoc"

```
var req = null;    // à déclarer en dehors pour être partagée entre les 2 fonctions

function submitForm() {    // fonction appelée par le client sur événement
    document.getElementById("zone").innerHTML = "Wait server...";
    if (window.XMLHttpRequest) req = new XMLHttpRequest();
    else if (window.ActiveXObject) req = new ActiveXObject(Microsoft.XMLHTTP);

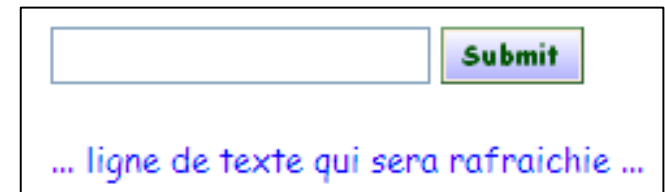
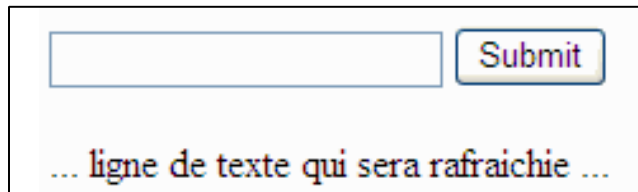
    req.onreadystatechange = mycallback;
    req.open("GET", "data.txt", true);    // false pour appel synchrone
    req.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    req.send(null);
}

function mycallback() {    // fonction appelée par le serveur en fin de traitement
    var res = null;
    if(req.readyState == 4) {
        if(req.status == 200) res = req.responseText + " " + document.ajax.texte.value;
        else res = "Error: returned status code " + req.status + " " + req.statusText;
        document.getElementById("zone").innerHTML = res;
    }
}
```

En bleu : manipulation de l'arbre DOM

Dynamisme des pages & plasticité

- Possibilité de manipuler l'arbre DOM durant l'exécution de l'application
 - **adaptation de la structure** : ajout/suppression/remplacement de bouts d'IHM à la volée
 - **adaptation du placement**
- Possibilité de modifier le CSS : **adaptation du style**



- ➔ Mise en œuvre partielle du levier de plasticité
« remodelage »

Gestion de l'indépendance vis à vis des navigateurs ou *plasticité aux navigateurs*

- Ajax complique les comportements habituels du navigateur (bookmark, enregistrement des pages, boutons précédent / suivant, référencement, etc)
 - Différence de prise en charge des navigateurs au niveau du DOM, Javascript, XMLHttpRequest, CSS, ...
 - Multiplicité des navigateurs et fonctionnement différent d'un OS à un autre
- ➔ Développement "cross-navigateur" fastidieux
Une problématique sans cesse renouvelée par les nouvelles versions des navigateurs

Gestion de l'indépendance vis à vis des navigateurs ou *plasticité aux navigateurs*

■ Les **librairies Ajax** viennent à la rescousse !

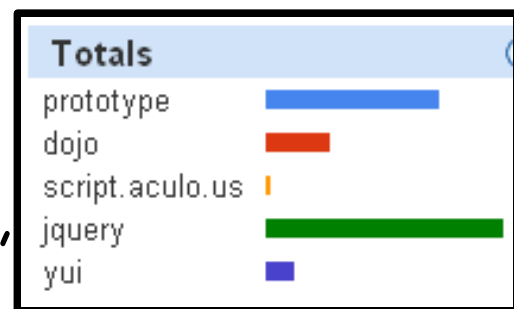
- gèrent pour nous les différences de navigateurs

■ Les librairies Ajax offrent aussi

- des widgets de haut niveau
- des effets visuels (surbrillance, slider, drag&drop, transitions visuelles : effets de flou, zoom, rotation, etc)
- des fonctions pour la manipulation DOM, la gestion des événements, la communication avec le serveur, l'internationalisation et l'historique, ...

■ Revient souvent à intégrer simplement un ou plusieurs fichiers javascript dans votre page

■ Exemples : Prototype, Dojo, Script.aculo.us, jQuery, Yahoo UI library, ...



Source : Google Insights

Hello World avec la librairie Dojo

Intégration de la librairie comme un simple "import" de fichier javascript

```
<html>
  <head>
    <script type="text/javascript"
      src="http://o.aolcdn.com/dojo/1.2.0/dojo/dojo.xd.js">
    </script>
    <script type="text/javascript">
      ...
      function submitForm() {
        ... // voir plus loin
      }
    </script>
    <style type="text/css">
      @import
      "http://o.aolcdn.com/dojo/1.2.0/dijit/themes/tundra/tundra.css";
    </style>
  </head>
  ...
```

Hello World avec la librairie Dojo

```
<body class="tundra">
  <FORM name="ajax" method="post" action="">
    <input type="text" size="20" id="texte"
      dojoType="dijit.form.TextBox" trim="true"/>
    <button id="submit" dojoType="dijit.form.Button">
      Submit
      <script type="dojo/method" event="onClick">
        dijit.byId("zone").setContent('Wait server...');
        submitForm();
      </script>
    </button>
  </FORM>
  <div id="zone" dojoType="dijit.layout.ContentPane">
    ... ligne de texte qui sera rafraichie ...
  </div>
</body>
</html>
```

Hello World avec la librairie Dojo

```
<script type="text/javascript">
    // importation des éléments de l'API Dojo requis
    dojo.require("dijit.form.Button");
    dojo.require("dijit.form.TextBox");
    dojo.require("dijit.layout.ContentPane");

    function submitForm() {
        dojo.xhrGet( { // équivalent du XMLHttpRequest
            url: "data.txt",
            handleAs: "text", // autre possibilité : xml
            load: function(data) {
                dijit.byId("zone").setContent(data + ' ' +
                                                dijit.byId("texte").getValue());
            },
            error: function(data, ioArgs) {
                alert("Error: returned status code " + ioArgs.xhr.status);
            }
        });
    }
</script>
```

Intérêts des librairies



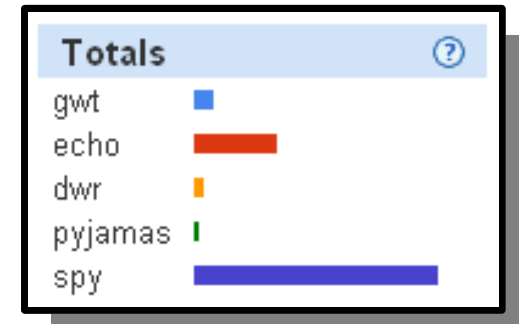
- Encapsulation communication via XMLHttpRequest
- Prise en charge de l'aspect cross-navigateur
=> **plasticité à la conception vis-à-vis des navigateurs !**
- Widgets de haut niveau



- Code JavaScript difficile à écrire
(peu d'experts)
- Code JavaScript compliqué à débbugger (alert !)
- Extensibilité ?

Frameworks AJAX

- Conservent les avantages des librairies tout en évitant généralement de manipuler directement le Javascript
- **Framework** = librairies + IDE et outils dérivés
- Exemples en Java :
GWT (Google Web Toolkit), Echo, DWR (Direct Web Remoting), ICEfaces (java+JSF tags), ...
- Exemples en .Net :
ASP.Net AJAX, Projet Volta (GWT-like), ...
- Exemples en Python :
Pyjamas (GWT-like)
- Exemple avec de l'HTML étendu :
Spry



Source : Google Insights

GWT : Développer, débbugger, tester en Java

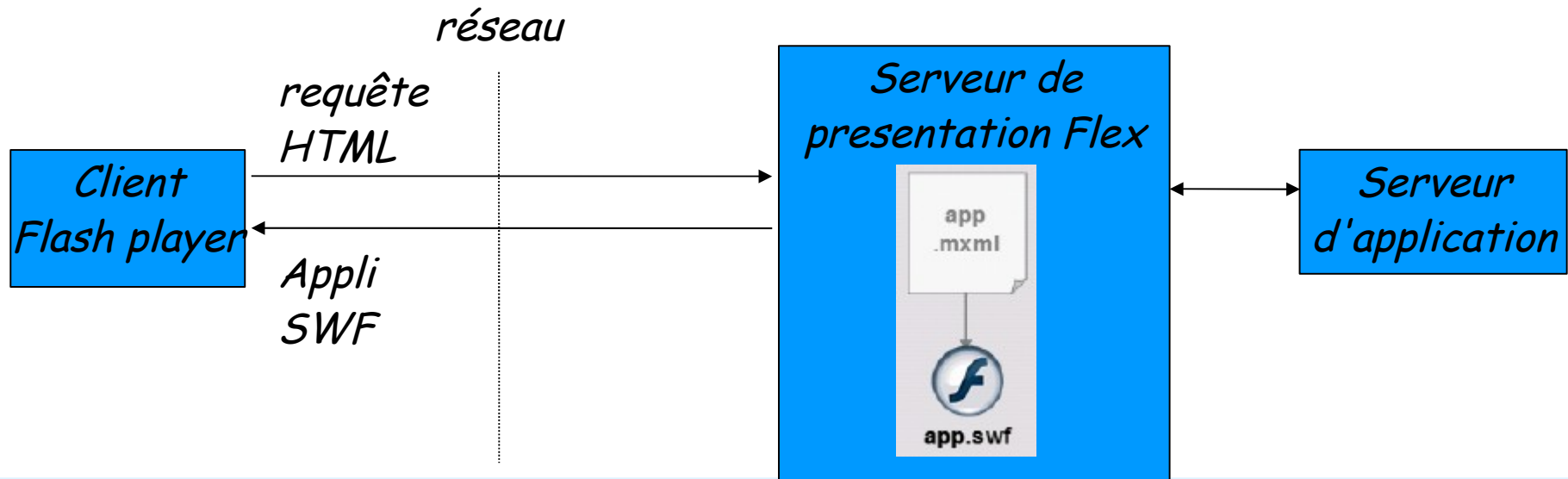
- Les IHM sont réalisées en Java (seulement un sous-ensemble)
 - Bibliothèque de widgets proche de Swing
 - Compilation vers Javascript/CSS
- Simplification de l'écriture d'applications cross navigateurs ...
- ... dans un environnement pour les développeurs
 - Typage fort de Java vs non typage de javascript
 - Support des IDE (mode debug, JUnit, ...)
 - Possibilité de composer graphiquement les interfaces (GWT designer, GWT editor)

Avantages & inconvénients de AJAX/GWT

- De manière générale :
 - L'option du navigateur JavaScript doit être activée sinon certaines widgets ne fonctionnent pas (bookmarks, boutons précédent/suivant, etc)
 - Des différences de comportement entre navigateurs si développement "artisanal"
- Spécifiquement à GWT :
 - Intégration facilitée avec les autres frameworks (Struts, Spring, J2EE etc.)
 - Support pour l'internationalisation et l'historique
 - Code généré difficile à debugger
 - Bibliothèque de widgets pas très riche

Flex en bref

- Application SWF (Shockwave Flash) intégrée à une page HTML
- Déploiement de la page HTML sur un serveur de présentation Flex
- Accès à la page par un navigateur via une URL comme une page HTML classique
- Téléchargement de l'appli SWF sur le client
- Accès à des services distants via des requêtes HTTP entre autre et échange de données en XML



FLEX = MXML + AS3 + RSL

■ MXML

- définit le positionnement des éléments d'IHM
- gère les événements
- fait la liaison entre les éléments d'IHM et les données
- définit l'accès aux sources de données serveur

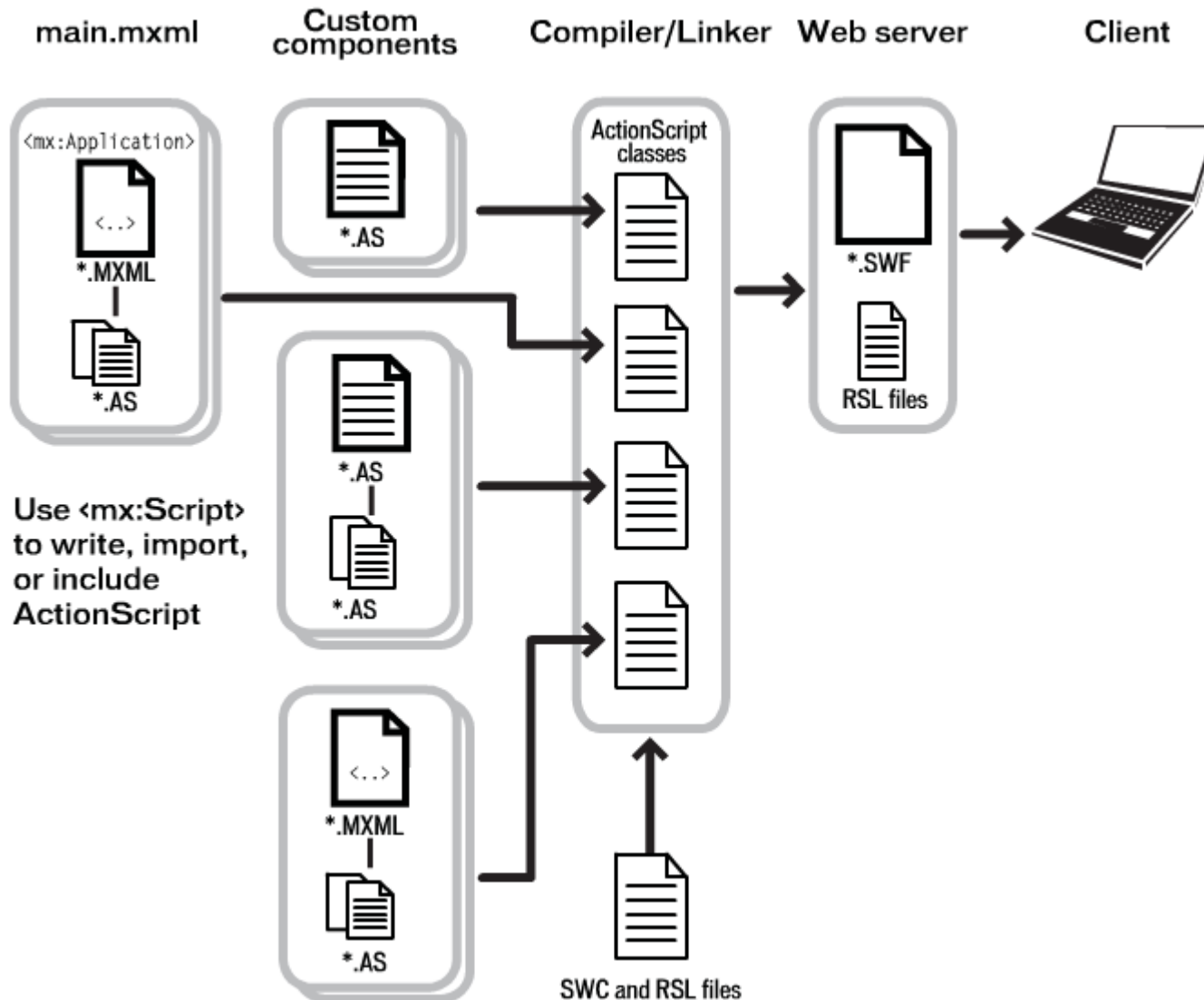
■ AS3 (ActionScript3)

- décrit la logique applicative nécessaire coté client
- supporte un équivalent de XMLHttpRequest

■ RSL (Runtime Shared Library)

- permet de partager des briques d'IHM, des ressources et des librairies entre plusieurs fichiers SWF

Compilation MXML/AS3 vers SWF



Source : Adobe ([flex3_buildanddeploy.pdf](#))

Hello world en Flex

- Description des éléments d'IHM et événements en MXML et traitement en AS3

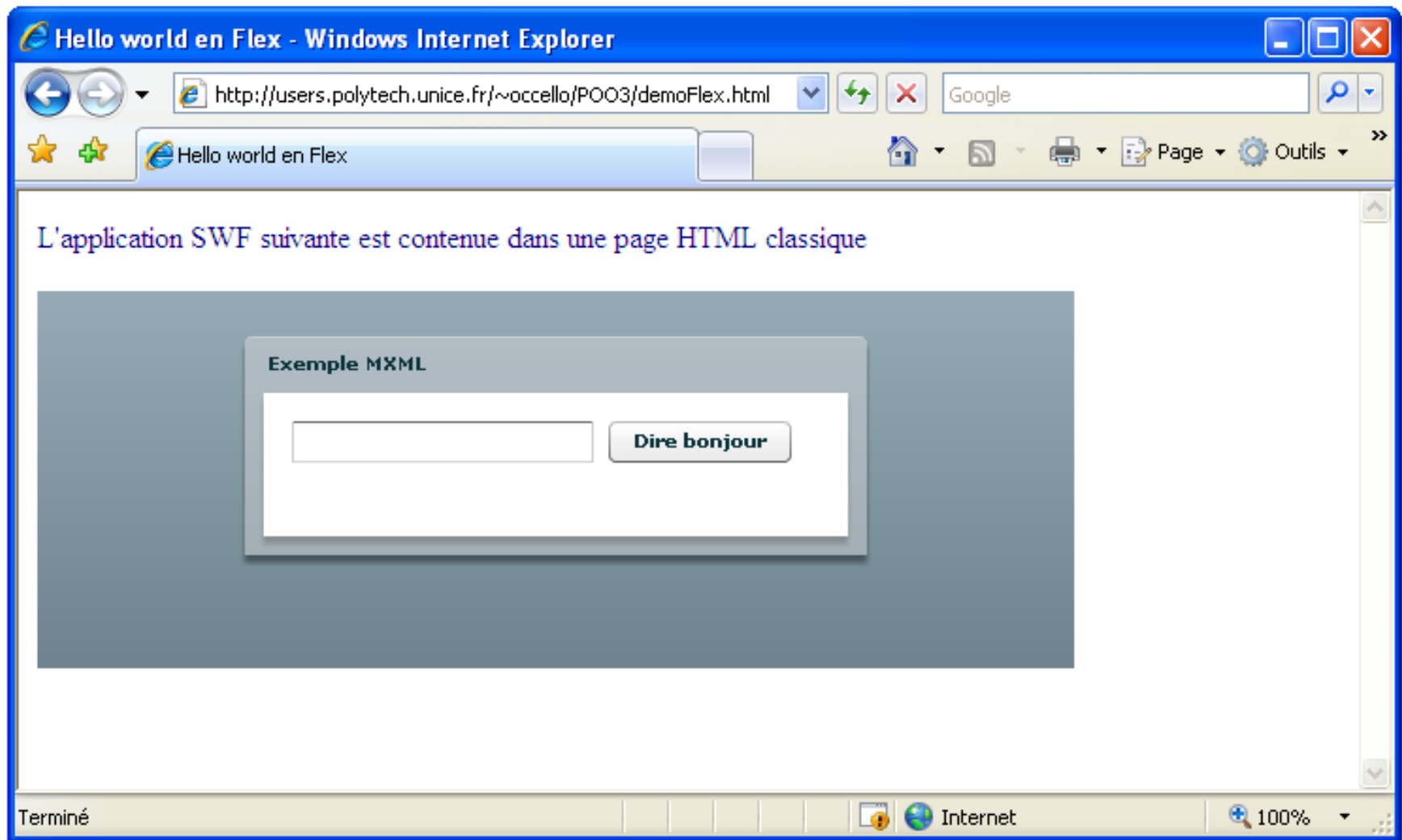
```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="vertical" horizontalAlign="center">
    <mx:Panel title="Exemple MXML" width="330"
        paddingLeft="15" paddingTop="15"
        paddingBottom="15" paddingRight="15">
        <mx:HBox>
            <mx:TextInput id="firstname"/>
            <mx:Button label="Dire bonjour"
                click="{result.text='Bonjour, '+firstname.text}"/>
        </mx:HBox>
        <mx:Label id="result" width="265"/>
    </mx:Panel>
</mx:Application>
```

Hello world en Flex

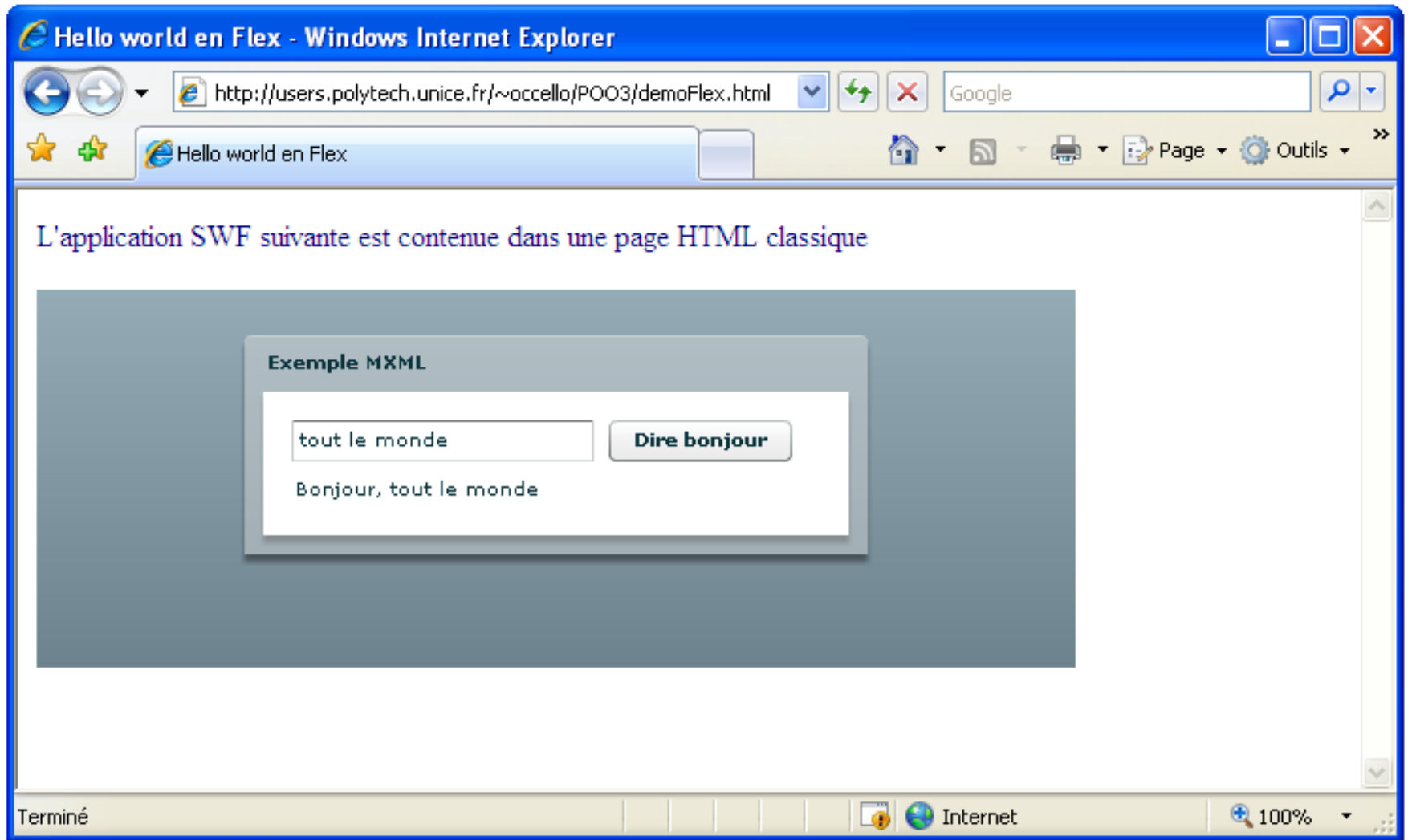
- Inclusion du fichier swf obtenu après compilation du MXML/AS3 dans une page web classique

```
<html>
  <head>
    <title> Hello world en Flex </title>
    <meta http-equiv="Content-Type"
          content="text/html; charset=UTF-8" />
  </head>
  <body text="#000080">
    <p>L'application SWF suivante est contenue dans une page HTML
      classique
    </p>
    <object>
      <embed src="HelloFlex.swf" width="550" height="200">
      </embed>
    </object>
  </body>
</html>
```

Hello world en Flex



Hello world en Flex



Outillage autour de Flex

- SDK gratuit : compilateur + bibliothèque composants graphiques + bibliothèque services applicatifs (glisser/déposer, animation, liaison des données; ect)
 - permet de générer une application SWF à partir de MXML/AS3
- Flash player (gratuit)
 - est le plugin à rajouter au navigateur pour interpréter le byte code d'une appli Flex
- Flex Builder (payant)
 - outil de développement, pas obligatoire
 - disponible en Standalone ou en plugin Eclipse
 - utile surtout pour débbugger et composer graphiquement les interfaces

Les avantages de Flex

- Rendu identique sur la plupart des navigateurs et des systèmes d'exploitation
- Pas besoin de gérer l'aspect "cross-navigateur", flash assure l'interopérabilité pour nous
- Bibliothèque de widgets riches et soignés
- Plus facile à apprendre que GWT ?

Comparatif des 3 concurrents majeurs

Points communs :

- Utilisation d'un langage à balises pour décrire la structure de l'IHM
- Séparation style / contenu de l'IHM
- Langage additionnel pour la partie traitement

	Ajax	Flex	Silverlight
IHM Style	CSS	MXML + Flex CSS	XAML + templates
IHM Structure	HTML	MXML	XAML
Logique applicative coté client	Javascript	Actionscript	En .Net

Comment choisir sa technologie RIA ?

- Choix difficile mais important
 - Difficile de migrer vers une autre technologie (pas de réutilisation)
 - Pas de plasticité à ce niveau
- Quelques critères de sélection à prendre + ou - en compte ...
 - Variété des widgets et des effets visuels
 - Spécificité des fonctionnalités avancées
 - Outillage associé et intégration avec d'autres techno
 - Importance de la communauté et qualité de la documentation
 - ...
- ... en fonction de vos besoins :
 - Taille de l'application à réaliser et domaine applicatif
 - Durée du développement (prototype vs production)
 - Compétences des développeurs et IDE utilisé
 - ...

Cas particulier de OpenLaszlo

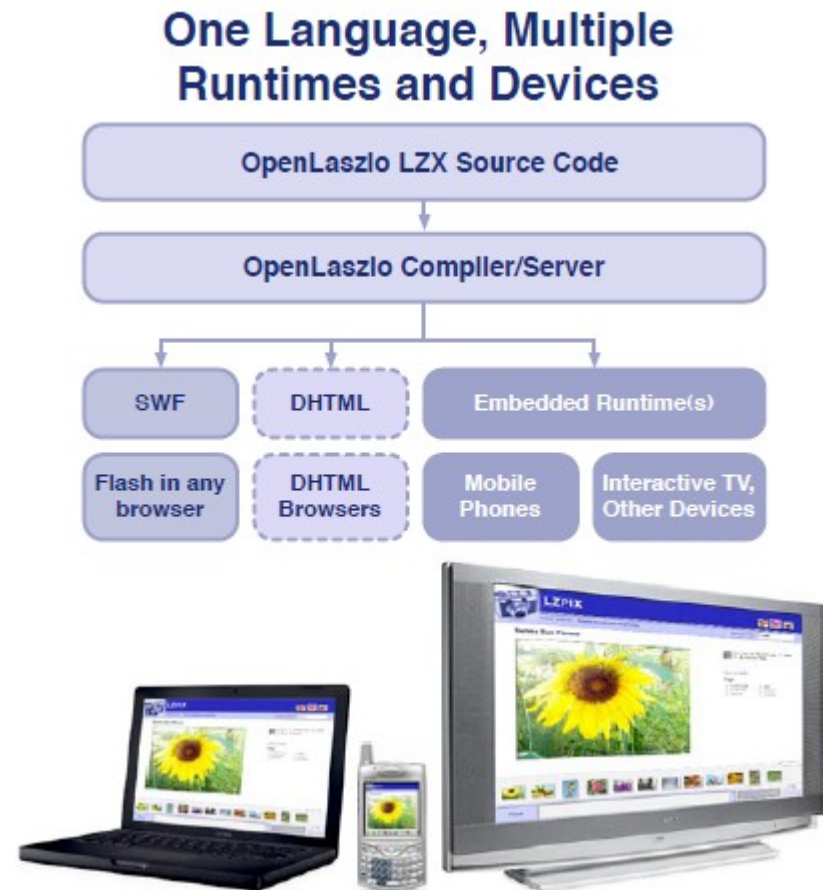
- Un langage à balises pour décrire la structure de l'IHM : LZX

- Génération de code vers des cibles multiples :

DHTML,
différentes versions de flash
et bien d'autres ...

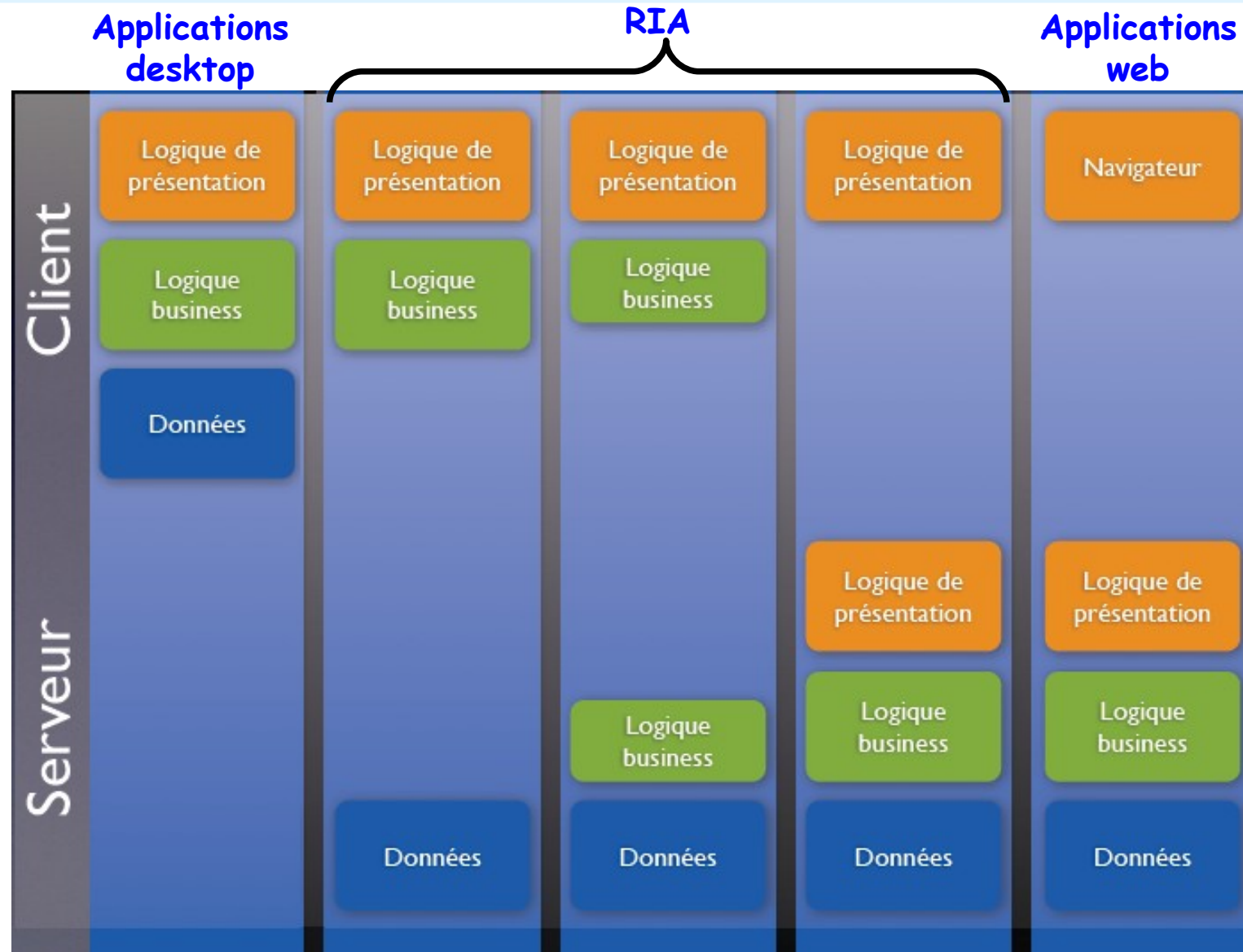
bientôt Java Swing

→ Plasticité à la conception
vis à vis des plates-formes
logicielles & matérielles



source: openlaszlo.org

Convergence des types d'applications



Source : <http://www.camilleroux.com/wp-content/uploads/2008/12/Cours-RIA-Flex.pdf>

Deux déclinaisons des RIA : RWA vs RDA

- RWA pour Rich Web Application

- Une évolution des applications web traditionnelles pour intégrer les avantages des applications "desktop" (en particulier expérience utilisateur)
- Proche des "clients légers"

- RDA pour Rich Desktop Application

- Une évolution des applications "desktop" pour intégrer les avantages des applications web (diffusion, maintenance)
- Proche des "clients lourds"

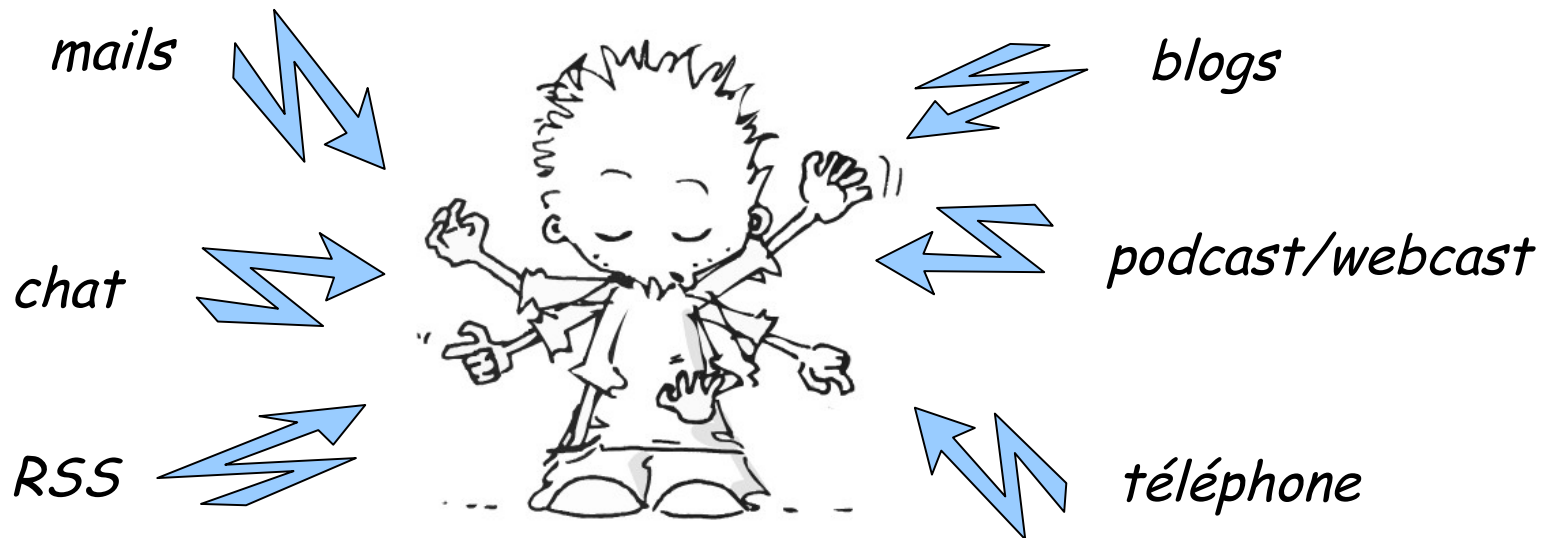
Ce qu'il faut retenir des RIAs

- RIA = le meilleur du web et du "desktop"
- RIA & conception des interfaces
 - Séparer présentation - logique - données
 - Briques d'IHM réutilisables
- Nécessité d'installer « des choses en plus » dans le navigateur et forte concurrence sur les technologies
 - Multiplication des technologies sur le poste de travail !
 - Perte de la portabilité !
- Quid de la sécurité
- Quid de l'accessibilité
- Quid du référencement
- ...

Vers un Web3.0 ?

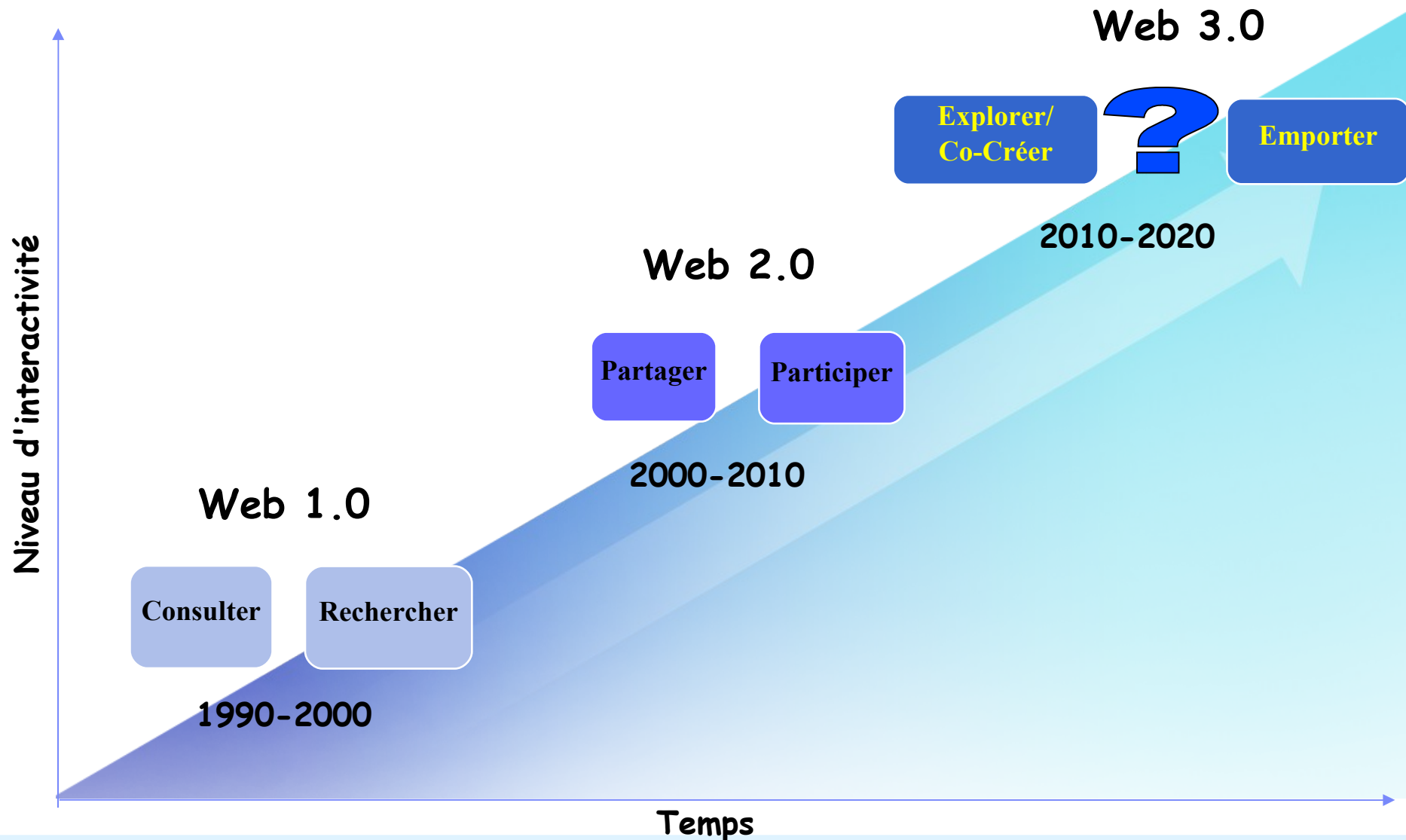
Une nouvelle vague d'évolution de Internet arrive déjà !

- D'avantage d'informations ont été produites ces dernières 30 années que durant les 5000 ans précédents !
 - Surcharge de l'information & des canaux de communication



Adapter les interfaces à deux aspects essentiels de l'être humain : notre nature innée sociale et visuelle

Quel sera le visage du Web3.0 ?

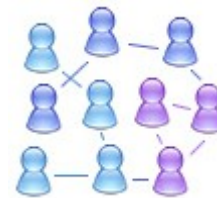


La piste de l'Internet collaboratif en 3D

- Aujourd'hui, environ 45 mondes virtuels regroupent près de 350 millions d'utilisateurs (source : K Zero)



- 1/5 des internautes sont dans un réseau social (source : Ifop, 2007)



- « By the end of 2011, 80 % of active Internet users (and Fortune 500 enterprises) will have a "second life", but not necessarily in Second Life » (source : Gartner Inc, avril 2007)

- Plus aller plus loin dans l'amélioration de l'expérience utilisateur
 - Ordinateurs + rapides, graphismes + réalistes, bande passante accrue
 - Des équipements à porter sur soi (wearable computer)
 - Des standards pour favoriser l'interopérabilité entre "les mondes"

La piste d'un Internet encore plus plastique

- D'autres leviers de plasticité existent et ne sont pas encore intégrés aux applications Internet

- Changement des modalités d'interaction

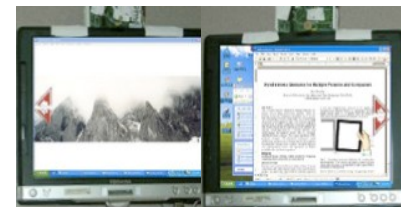


- Migration de l'IHM vers un autre support



- Distribution de l'IHM entre plusieurs supports

- ...



- Vers des RIA++ ?

3D + plasticité

Convergence vers une réalité augmentée ?

Quelques liens intéressants ...

- *Actualité des RIAs*
<http://www.riapedia.com/>
- *Tutoriaux et demos sur Ajax*
<http://www.xul.fr/xml-ajax.html>
- *Des cours en ligne autour des techno Ajax*
<http://www.javapassion.com/ajaxcodecamp/>
- *Tutoriaux sur Flex*
<http://www.flex-tutorial.fr/>
- *Librairie de briques d'IHM Flex gratuits*
<http://code.google.com/p/flexlib/wiki/ComponentList>
- *RIA & utilisabilité*
http://www.digital-web.com/articles/usability_for_rich_internet_applications/
<http://www.nielsbruin.nl/blog/?p=162>
http://www.boxesandarrows.com/view/usability_heuristics_for_rich_internet_applications
<http://www.nngroup.com/reports/flash/RIA-usability.pdf>
- *RIA & Accessibilité*
<http://www.w3.org/WAI/intro/aria>
<http://my.adobe.acrobat.com/riaaccess>