

Université de Msila
Faculté des mathématiques et de l'informatique
Département d'informatique

3^{ième} année Licence ISIL

Introduction aux applications Web

Présenté par : Meliouh.A
2019-2020

PLAN DU COURS

- 1. Introduction**
- 2. Généralités sur le Web**
- 3. Historique et évolution du Web**
- 4. Difficultés du développement Web**
- 5. Le Web dans le monde actuel**
- 6. Les applications web**

1. Introduction

- **Module Complet** : 45h
- **Crédits** : 4
- **Coefficient** : 2
- **Objectifs de l'enseignement**
Apprendre a développer des application Web avancées.
- **Connaissances préalables recommandées**
Les connaissances fondamentales déjà acquises en Technologies Web.
- **Contenu de la matière:**
 - 1.Introduction aux applications Web
 - 2.Rappels sur les sites Web
 3. Notion d'architecture Web
 - 4.Introduction AJAX
 5. Javascript
 6. Développement web : mise en pratique

2. Historique et évolution du Web

1. Naissance du World Wide Web

Le Web est né au CERN en 1989-1990.

- Tim Berners-Lee a défini une architecture pour accéder à des documents liés entre eux et situés sur des serveurs reliés par Internet (Web = toile d'araignée)
- le W3C (World Wide Web Consortium) mis en place rapidement (1994) pour définir des standards (ouverts).
- Accès à des documents structurés via des liens hypertextes
- Protocoles et langages simples
- Technologies de base HTML, HTTP, MIME, formats GIF. . .

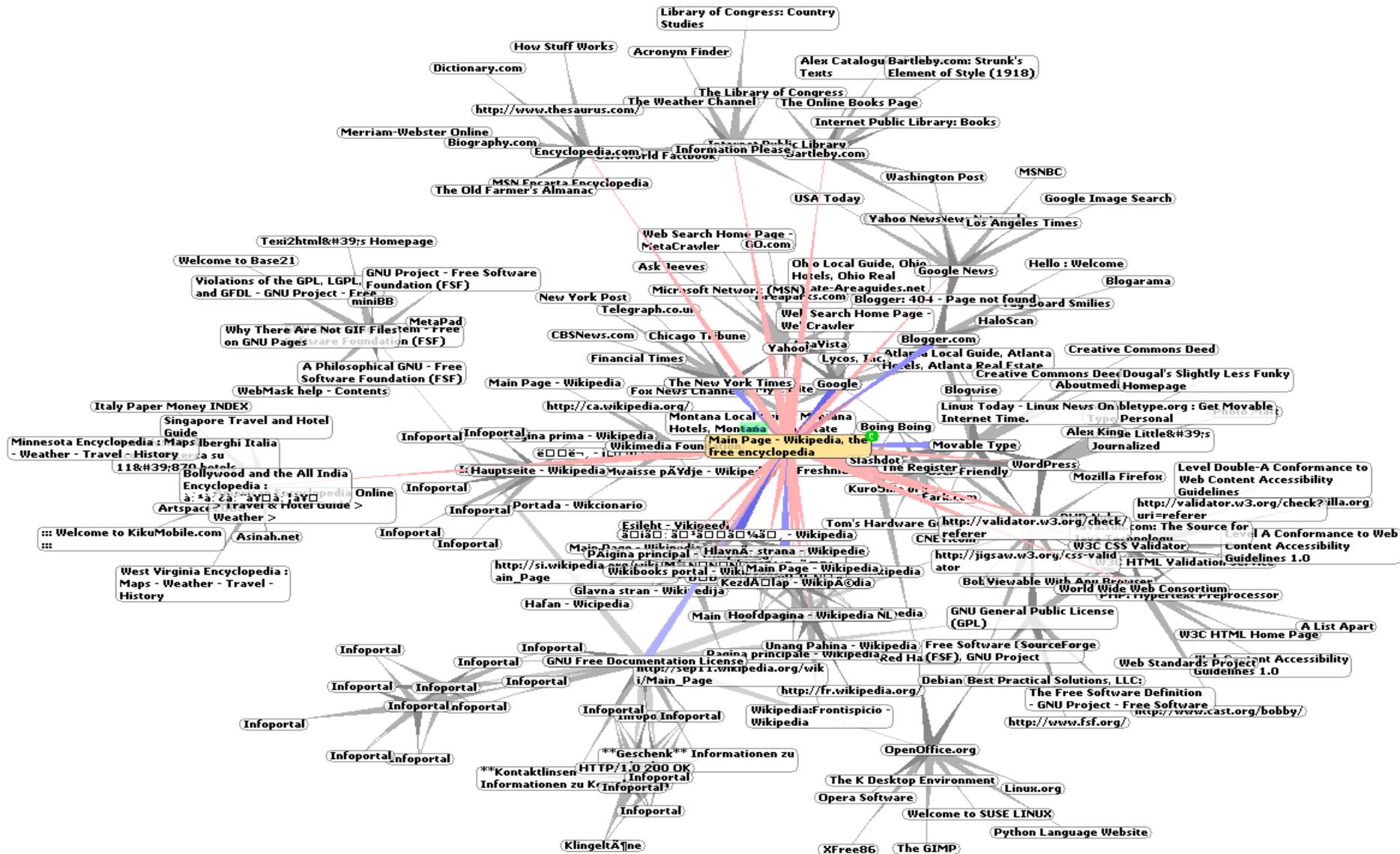
2. Ouverture, homogénéisation et programmation (fin des années 1990)

- Interactions avec les applications et programmation Web
- Langages plus riches, manipulation d'objets, développement des styles
- Evolution des technologies : XML, CSS, DOM, Server Pages, JavaScript . . .
- Standardisation difficile (guerre des navigateurs)

3. Evolution des usages et de l'interface utilisateur (depuis 2005)

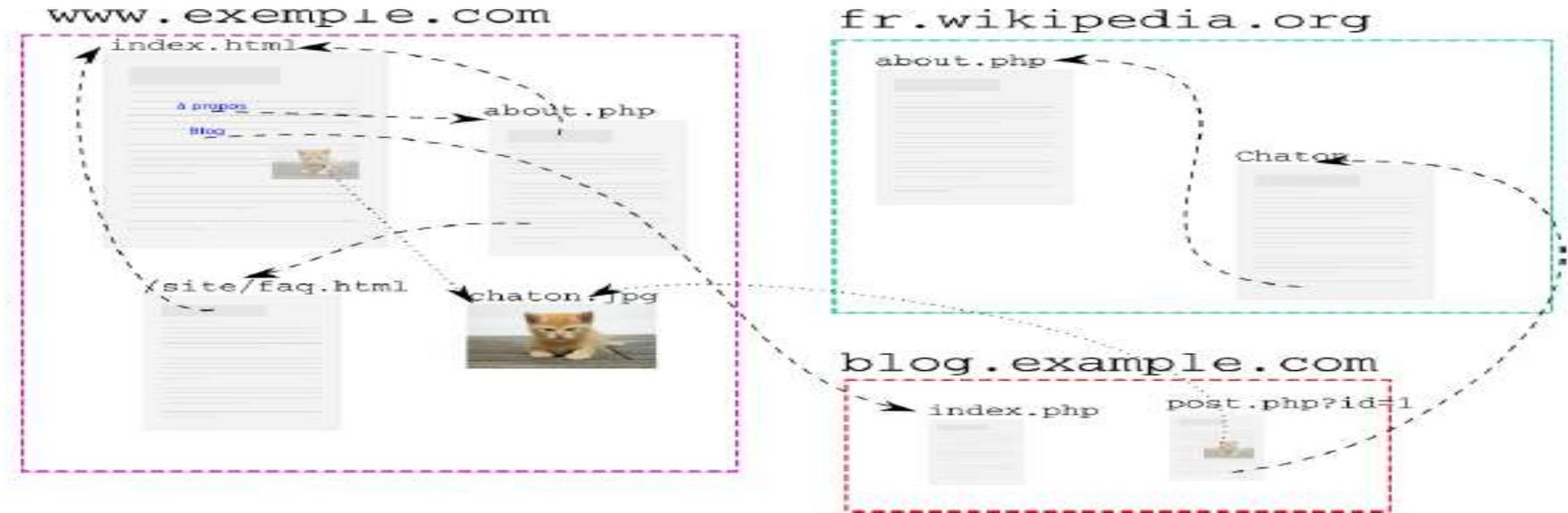
- Partage d'informations, édition collaborative, sites communautaires
- Réseaux sociaux, mondes virtuels
- Technologie AJAX, HTML 5
- Intégration de flux RSS, de vidéos, ...
- Personnalisation des accès
- User-generated content

3. Généralités sur le Web



3. Généralités sur le Web

➤ Graphe de ressources liées décentralisé



- Liens entre différents documents, du même site ou de sites différents.
- Certains liens sont navigables, d'autres correspondent à l'inclusion de ressources externes (images).

Note : les liens sont uni-directionnels, du point de vue des serveurs : les documents ne savent pas qui leur pointe dessus. Aucune coordination : pas de permission à demander avant de tisser des liens, mais pas de garantie de disponibilité des ressources distantes.

3. Généralités sur le Web

- **Ressources**

- Documents (statiques)

- Consultation dynamique :

- ressources issues d'applications (dynamiques)

- représentation de ces ressources dans les navigateurs Web

- Éléments « virtuels » décrivant des « faits » (Web sémantique)

- **Ressources liées**

- Identification des ressources

- Localisation des ressources

- Agréger des ressources et leurs donner des propriétés pour créer de la connaissance

- **Décentralisé**

- Lier entre-elles des ressources localisées physiquement n'importe où sur Internet Confiance, provenance ?

3. Généralités sur le Web

➤ Comment ça marche ?



Les fichiers sont stockés sur des **serveurs Web**

Des clients (**les navigateurs Web**) se connectent au serveur en utilisant le protocole **HTTP** (protocole d'application au dessus de TCP/IP).

Les ressources sont identifiées par des **URLs** (des chaînes de caractères au format :
protocol://machine:port/chemin/vers/la/ressource).

3. Généralités sur le Web

➤ Programmation web client/serveur

Lorsqu'on connecte les ordinateurs en réseau, il devient intéressant de concentrer certaines ressources (c'est-à-dire des informations et des programmes) sur un seul ordinateur, et de permettre aux autres ordinateurs de se servir de ces ressources uniquement lorsqu'ils en ont besoin. C'est l'architecture client-serveur.

Le serveur, c'est l'ordinateur sur lequel se trouve une ressource.

Le client, c'est l'ordinateur qui a le droit d'accès à la ressource sur le serveur

➤ Côté serveur

- Exécution réalisée sur le serveur (PHP, Servlet)
- Résultat de l'exécution : page HTML envoyée par le serveur au navigateur

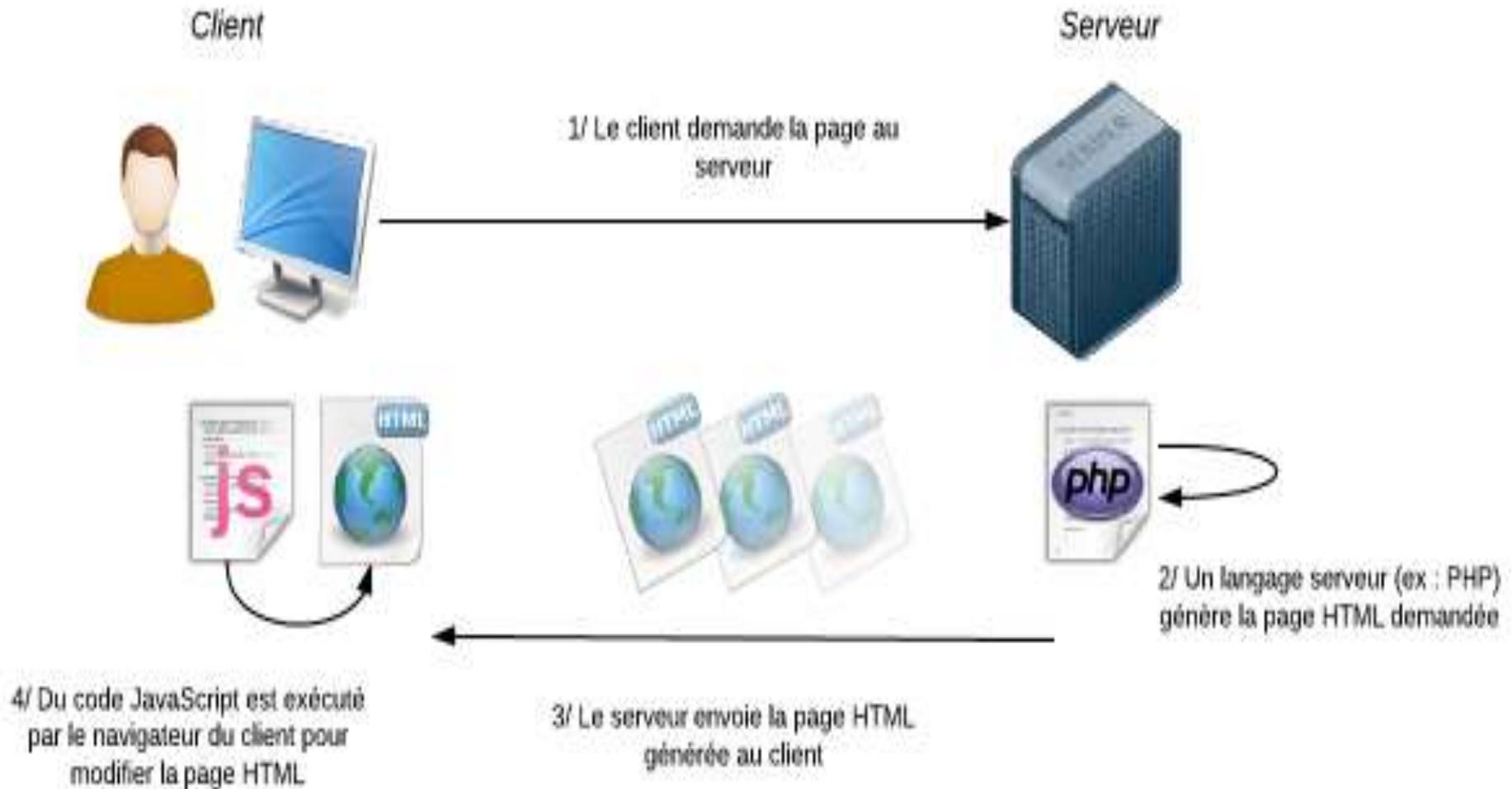
➤ Côté client

- Exécution réalisée sur le client
- Navigateur capable de réaliser l'exécution
- Transfert du code EMBARQUÉS dans la page HTML, depuis le serveur vers le client (HTML-embedded scripting)

Exemples : Scripts Javascript, Applets Java, ActiveX, Ajax

3. Généralités sur le Web

➤ Exemple

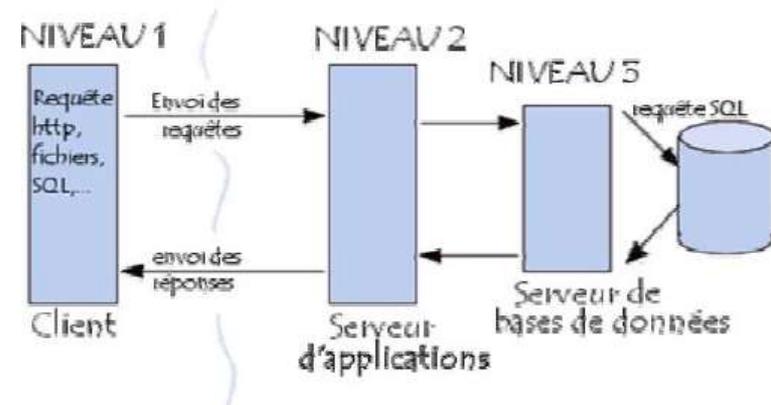
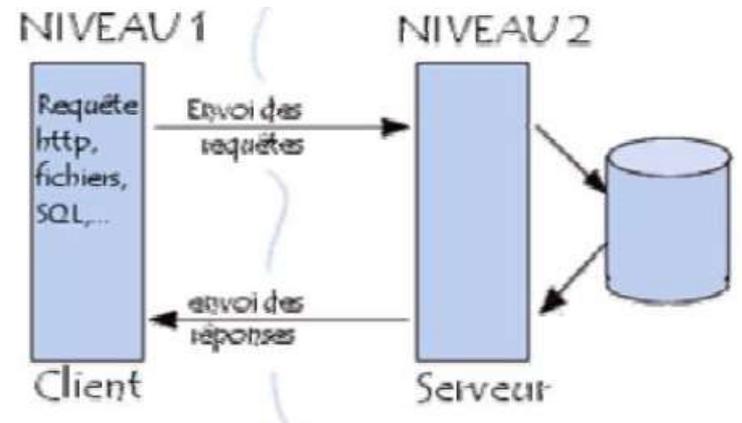


3. Généralités sur le Web

➤ L'Architecture N-tier

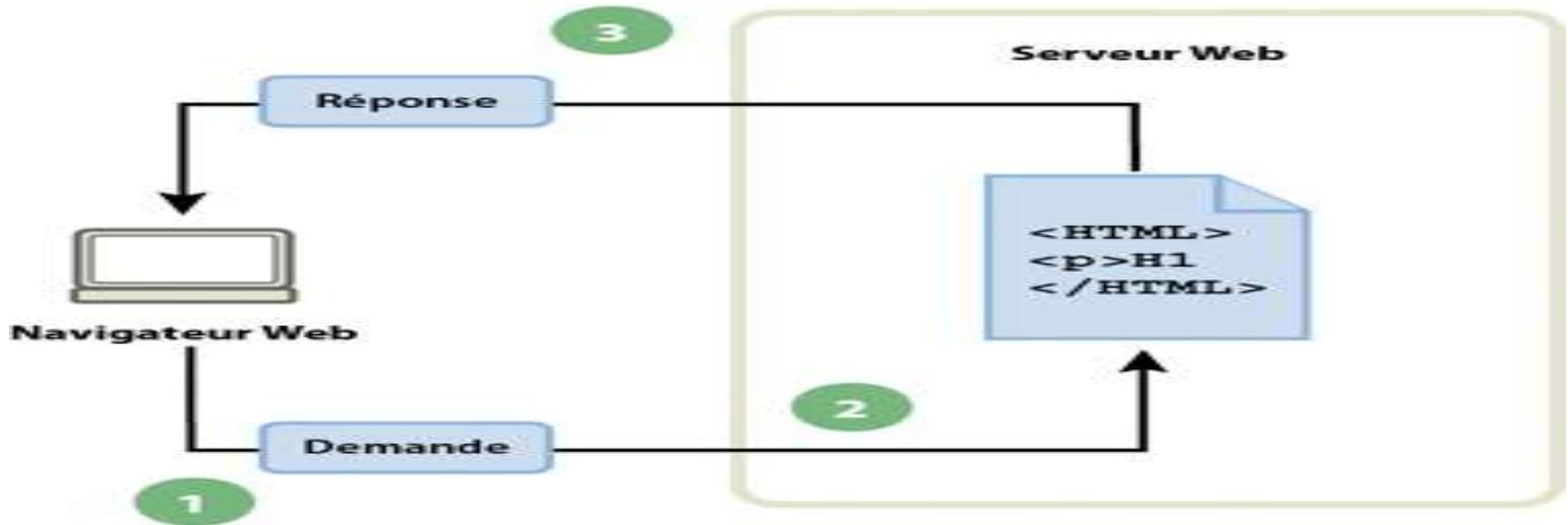
• **Tier** n'est surtout pas à interpréter comme 1/3, Il s'agit d'une utilisation du terme anglais **tier**, qui signifie **niveau**.

- L'architecture **client/serveur** possède deux types d'ordinateurs sur un réseau : les clients et les serveurs, elle possède donc deux niveaux et s'appelle **two-tier** en anglais.
- Les architectures **N-tier**, ou architectures distribuées, scindent le serveur en plusieurs entités. Par exemple, un serveur d'application qui lui-même est le client d'un serveur de base de données.



3. Généralités sur le Web

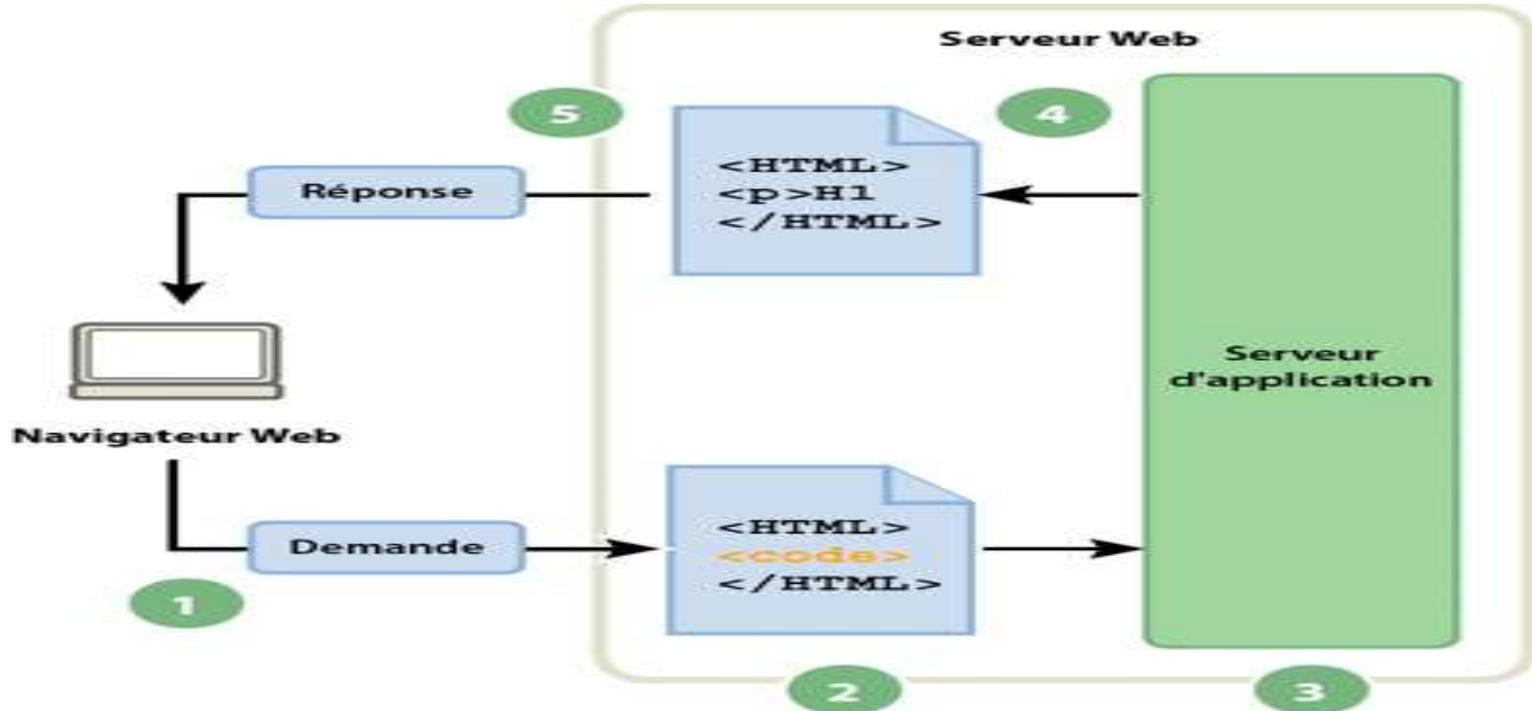
➤ Site Web statique



1. Le navigateur Web demande une page statique.
2. Le serveur Web recherche la page.
3. Le serveur Web envoie la page au navigateur requérant

3. Généralités sur le Web

➤ Site Web dynamique Architecture 2-tiers

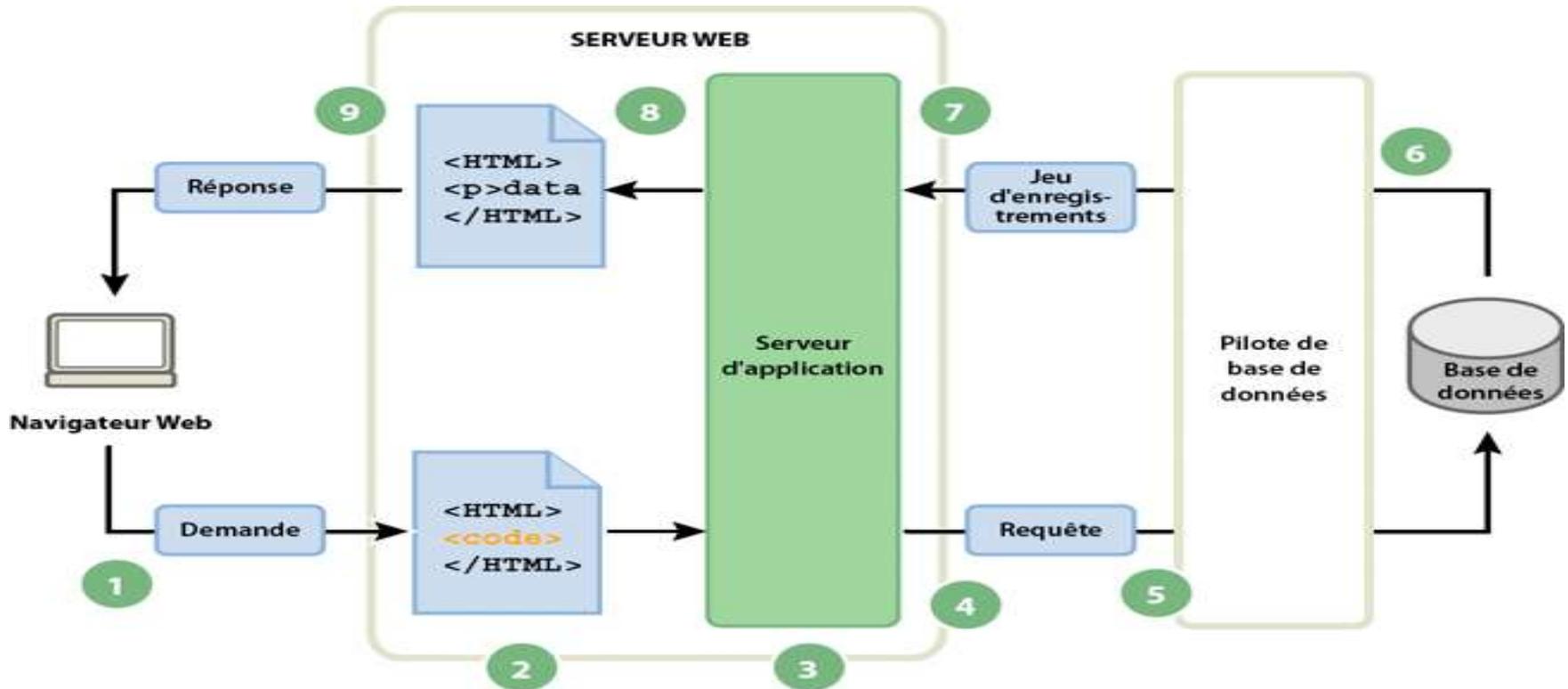


1. Le navigateur Web demande une page dynamique.
2. Le serveur Web recherche la page et la transmet au serveur d'application.
3. Le serveur d'application analyse la page à la recherche d'instructions et la termine.
4. Le serveur d'application transmet la page terminée au serveur Web.
5. Le serveur Web envoie la page achevée au navigateur requérant.

3. Généralités sur le Web

➤ Site Web dynamique

Architecture 3-tiers



6. Le jeu d'enregistrements est renvoyé au pilote de la BD.
7. Le SGBD transmet le jeu d'enregistrements au serveur d'application.
8. Le serveur d'application insère les données dans la page, puis transmet la page au serveur Web.
9. Le serveur Web envoie la page achevée au navigateur requérant.

3. Généralités sur le Web

➤ Protocole HTTP

- Le Protocole HTTP (HyperText Transfert Protocol) sert notamment au dialogue entre un client web (navigateur par exemple) et un serveur (apache par exemple).
- Comme la plupart des protocoles de la couche Application, c'est un protocole orienté texte (ASCII), basé sur TCP. Il existe deux spécifications la 1.0 et la 1.1 (RFC 1945).



3. Généralités sur le Web

➤ L'adressage web : URI/URL

- Un URI (Uniform Resource Identifier) soit littéralement « identifiant uniforme de ressource », est une courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web.

- Un URL (Uniform Resource Locator) littéralement « localisateur uniforme de ressource », est une chaîne de caractères utilisée pour adresser les ressources du World Wide Web : document HTML, image, son, forum Usenet, boîte aux lettres électronique, etc. Elle est aussi appelée une adresse web.

- Le format d'une adresse web ou URL :

protocole://[<login>:<mot_de_passe>]<serveur>[:<numero_port>]/[<chemin>]/<ressource>

Exemples : **http://www.example.com/tim/page.html**

http://192.168.52.83/index.php

ftp://ftp.is.co.za:21/rfc/rfc1808.txt

4. Difficulté du développement Web

➤ Compétences

- Il faut maîtriser de nombreuses technologies :
 - structure du document : XML, XHTML, DOM
 - rendu : CSS (feuilles de style)
 - interaction : langage coté client (Javascript)
 - interaction : langage coté serveur (PHP, Perl, Python, ...)
 - XSLT pour l'échange d'information
 - les frameworks :
 - PHP : Pear, Zend
 - Javascript : prototype, JQuery, Rico, Scriptaculous
 - XML : JaSON, YML
 - les CMS (Content Management Systems) : Joomla, SPIP,
 - ...
 - les Servlets de Java, JSP, JSP-EL, JSF
- Difficultés supplémentaires
 - ces technologies sont en constante évolution
 - de nouvelles technologies apparaissent tous les ans

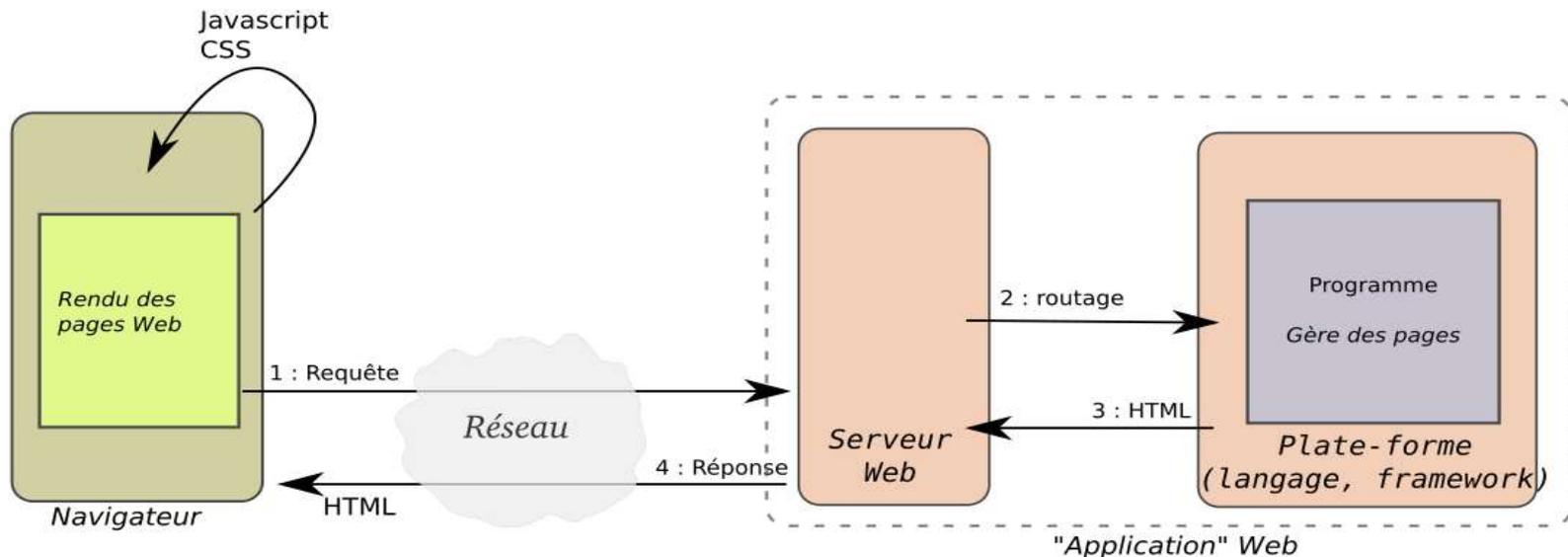
5. Le Web dans le monde actuel

➤ Aspects économiques et psychologiques

- Internet et le World Wide Web sont devenus un formidable vecteur d'information. Tant au niveau de la diffusion (broadcasting) que du partage d'information et de la coopération (ex. wikipedia).
- Aujourd'hui, le site web représente la vitrine d'une entreprise
- Un site bien conçu représente donc un avantage certain pour l'entreprise et est le gage de son sérieux et de sa qualité

6. Applications Web

- Une **application web** désigne un logiciel applicatif (logiciel métier) hébergé sur un serveur et accessible via un navigateur web.
- Les technologies utilisées pour développer les applications web sont les **mêmes** que celles employées dans la **création des sites dynamiques**.



6. Applications Web

➤ Exemples d'applications Web

- Un moteur de recherche
- Un Webmail
- Un système de gestion de contenu
- Un web log
- Un wiki
- Un site web marchand
- ventes aux enchères.
- Un jeu par navigateur
- Facebook
- Un logiciel de forum
- etc...

6. Applications Web

➤ Les Clients

Client lourd « fat client, heavy client, thik client »

- C'est un logiciel exécuté sur le système d'exploitation de l'utilisateur et qui propose des fonctionnalités complexes avec un traitement **autonome**.
- Demande un effort de développement et tend à mêler la logique de présentation (l'interface graphique) avec la logique applicative (les traitements).
- Une nouvelle version doit être installée afin de la faire évoluer.
- En terme réseau, tout terminal équipé pour stocker et traiter par **lui-même** les données est considéré comme un client lourd.
- le client lourd ne dépend du serveur que pour l'échange des données dont il prend généralement en charge l'intégralité du traitement.

6. Applications Web

➤ Les Clients

Client léger « thin client »

- Le terme de client léger décrit les logiciels qui utilisent le modèle client-serveur, dans lequel toutes les opérations de traitement sont effectuées par le serveur.
- En www, cela désigne une application accessible via une **interface web** basique, consultable à l'aide d'un **navigateur web**, où la totalité du traitement est du côté du **serveur**.
- Le fait que l'essentiel des traitements soit réalisé du côté du serveur et que l'interface graphique est envoyée au navigateur à chaque requête permet **une grande souplesse de mise à jour**.
- En contrepartie, l'application doit s'affranchir des différences d'interprétation du code HTML/CSS par les différents navigateurs et l'ergonomie de l'application possède un champ réduit.
- En terme réseau, tout terminal qui ne sert que d'interface d'affichage et de saisie est un client léger.

6. Applications Web

➤ Les Clients

Intérêt du client léger

- Dans le cadre des applications « Web », l'utilisation du client léger simplifie le travail en éliminant le besoin de diffuser, puis d'installer un logiciel client sur les machines des utilisateurs,

Inconvénient du client léger

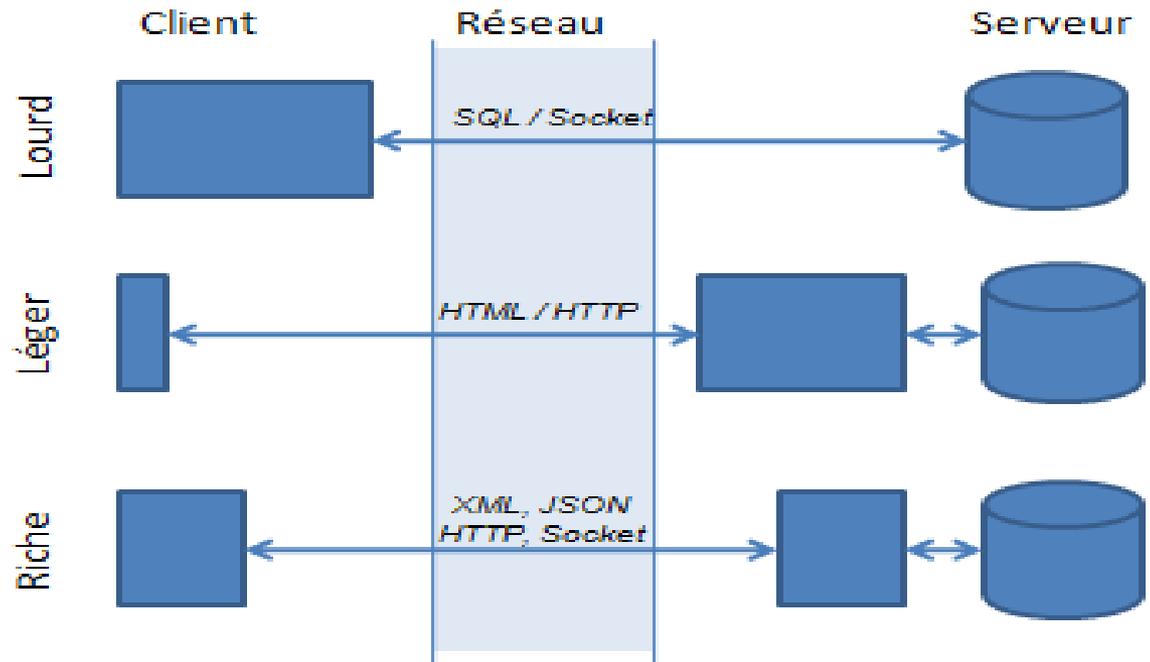
- Manque d'interaction avec les outils du bureau (en particulier l'Office suite)
- L'utilisation de clients légers utilise beaucoup de bande passante.
- En cas d'indisponibilité du réseau, plus aucune application n'est utilisable
- **Limites ergonomiques** : pour des raisons de sécurité, les technologies Web n'ont qu'un accès très limité aux ressources système du poste de travail.
- **Performances** : le serveur se chargeant de calculer l'ensemble des pages pour tous les utilisateurs, les performances de l'application peuvent être très dégradé lorsqu'un nombre important d'utilisateurs sont connectés.

6. Applications Web

➤ Client Riche « Rich Client »

- Compromis entre le client léger et le client lourd.
- C'est un programme qui permet d'afficher l'interface utilisateur d'une application web, avec cette particularité que l'interface utilisateur est **riche**, avec un comportement et une ergonomie identique à ce que l'on trouve dans un client lourd : **glisser déposer, onglets, multi fenêtrage, menus déroulants, mise à jour dynamique, ...**
- le **but** recherché est de disposer d'une plus grande ergonomie : **Rich User Interface**

- Aussi, L'intelligence logicielle est **répartie** équitablement entre le client et le serveur.



6. Applications Web

➤ Les Clients Riches

Les différentes technologies

- **XAML** (eXtensible Application Markup Language), prononcé « zammel », un standard XML proposé par Microsoft, utilisé notamment dans les applications utilisant le framework .NET
- **XUL**, prononcé « zoul », un standard XML proposé par la fondation Mozilla pour la description d'interface graphique. , utilisé par exemple dans le client de messagerie Mozilla Thunderbird ou dans le navigateur Mozilla Firefox.
- **Flex**, un standard XML proposé par la société Adobe.
- **FLASH** (Adobe),
- **Silverlight** (Microsoft)
- **Ajax** (XML, HTML et javascript), **(sera étudié avec + de détails)**
- **HTML5**,

6. Applications Web

➤ Les Clients Riches

Avantages du Client Riche

- **Ergonomie :**
 - interactivité plus riche,
 - plus grande expérience utilisateur permise par l'utilisation d'une machine virtuelle et de composants graphique (VS HTML).
 - actualisation par zone,
 - interfaces adaptées,
- **Performance :**
 - distributivité
 - diminution de la charge coté serveur,
 - limitation du flux réseau.
- **Déploiement :** déploiement à travers le navigateur Web.
- **Compatibilité :** multi plates-formes
- maintenance plus aisée,

6. Applications Web

➤ Les Clients Riches

inconvénients du Client Riche

- **Indépendance** : ces solutions étant propriétaires la maintenance et les évolutions des machines virtuelles correspondantes dépendent des Road Map des éditeurs concernées. Il est possible par exemple, qu'un jour Microsoft décide d'abandonner la plateforme Silverlight.
- **Licences** : ces solutions étant propriétaires il est nécessaire d'acquérir des licences à minima pour le studio de développement.
- **Langages de développement** : ces solutions ayant été pensée sur la base de solution graphique (le Flash pour Flex) les langages de développement varient beaucoup des langages pour client lourd ou léger.

En tant que site Web :

- Pas d'historique de navigation
- Pas de bookmark
- Problème d'indexation.
- Problème d'accessibilité

6. Applications Web

➤ Autres Clients

Remarque : Selon la complexité du terminal client et de son indépendance vis-à-vis du serveur il existe de clients : **super-léger, semi-léger, mi-lourd, super-lourd,...**

