Cycle PeiP
2ème année

POLYTECH MARSEILLE
Aix*Marseille Université

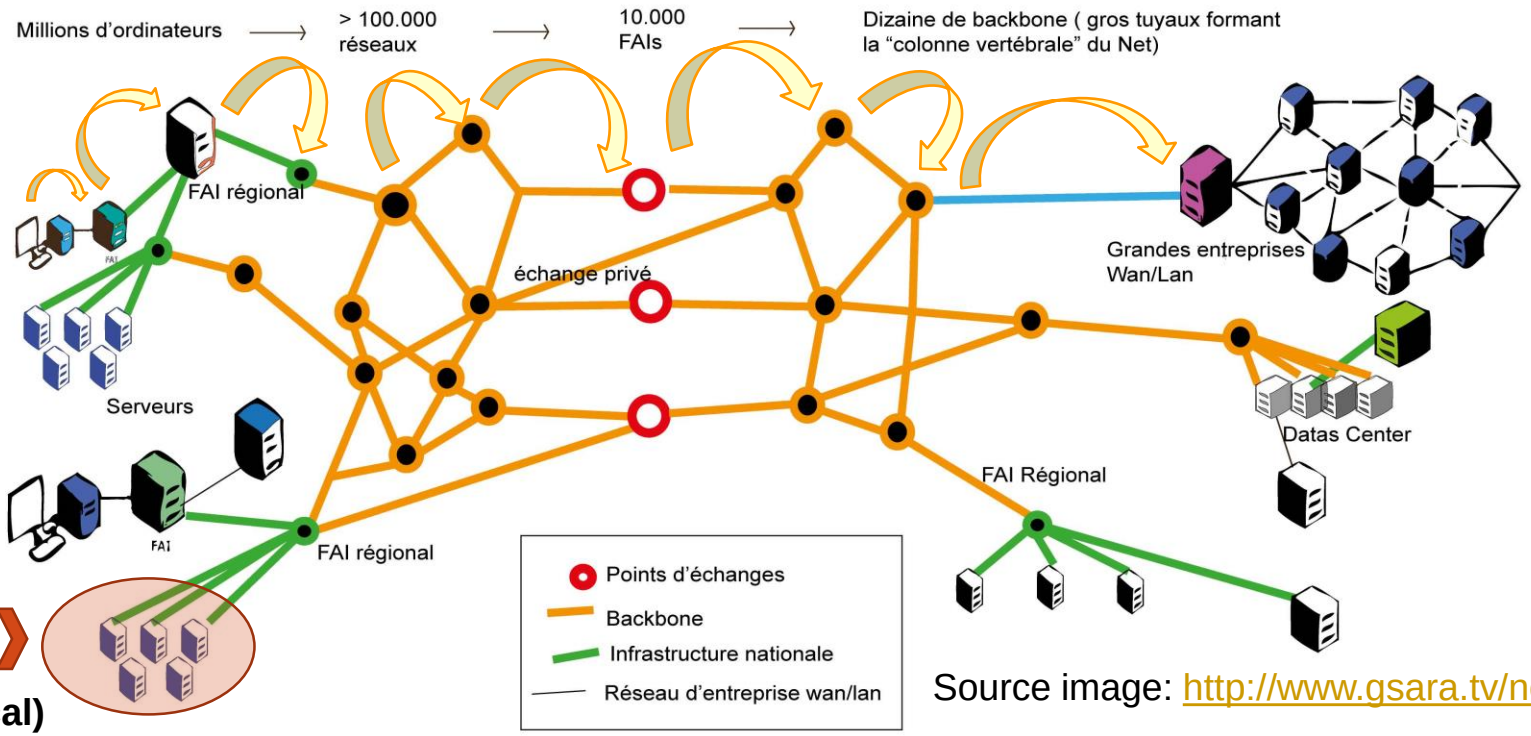
INTRODUCTION AU DEVELOPPEMENT « WEB »

Erwan TRANVOUEZ
erwan.tranvouez@univ-amu.fr
http://erwan.tranvouez.free.fr
2016 & before : P. Bellot & L. Hennoque
https://ametice.univ-amu.fr/course/view.php?id=49833

INTERNET ? *WHY* …

- Pourquoi ce cours ?
  - … pour des PeiP2 qui ne veulent pas faire de l'informatique
  1. **Parce que vous en ferez que vous le vouliez ou non …**
     - Dans la filière ingénieur que vous « intregrez » (C, Java, Basic, Fortran, python…)
     - Dans votre activité professionnelle
     - Dans vos loisirs (I know, vous le faites déjà)
  2. Parce qu'en tant qu'ingénieur, il est important (en terme de compétence voire de crédibilité) de savoir « grosso modo » d'où vient un problème… non pas pour le résoudre mais pour trouver la bonne personne qui résoudra le problème plus rapidement
  3. Et d'autres arguments taboux
- Genre de compétences :
  - Savoir si le problème vient du côté client ou du côté serveur, OS/logiciel ou matériel

INTERNET ! = RÉSEAU DE RÉSEAU...



LAN : Local Area Network (réseau local)

Source image: http://www.gsara.tv/neutralite/

Internet = WAN : Wide Area Network
=> réseaux locaux connectés

Internet est un réseau de réseaux d'ordinateurs connectés entre eux.
Les liaisons entre les ordinateurs constituent des « tubes » par lesquels transitent les données.

=> Illustration avec traceroute

DONC UN RÉSEAU C'EST QUOI ?

- Des machines qui cherchent à se connecter à d'autres machines pour demander des informations (éventuellement dans les 2 sens).
- Ceci implique de définir :
  - Les rôles de chaque machine (demandeur/offreur)
  - Comment ces machines communiquent :
    - En terme de ***transport « logistique »*** de l'information (adressage, comment rédiger une adresse, format de l'enveloppe)
    - En terme de ***contenu du message*** (forme et structure du contenu)
    - Eventuellement en terme de ***séquence de messages*** attendus (ex. accusé réception) => protocole

* Allégorie postale

INTERNET ≠ WEB…

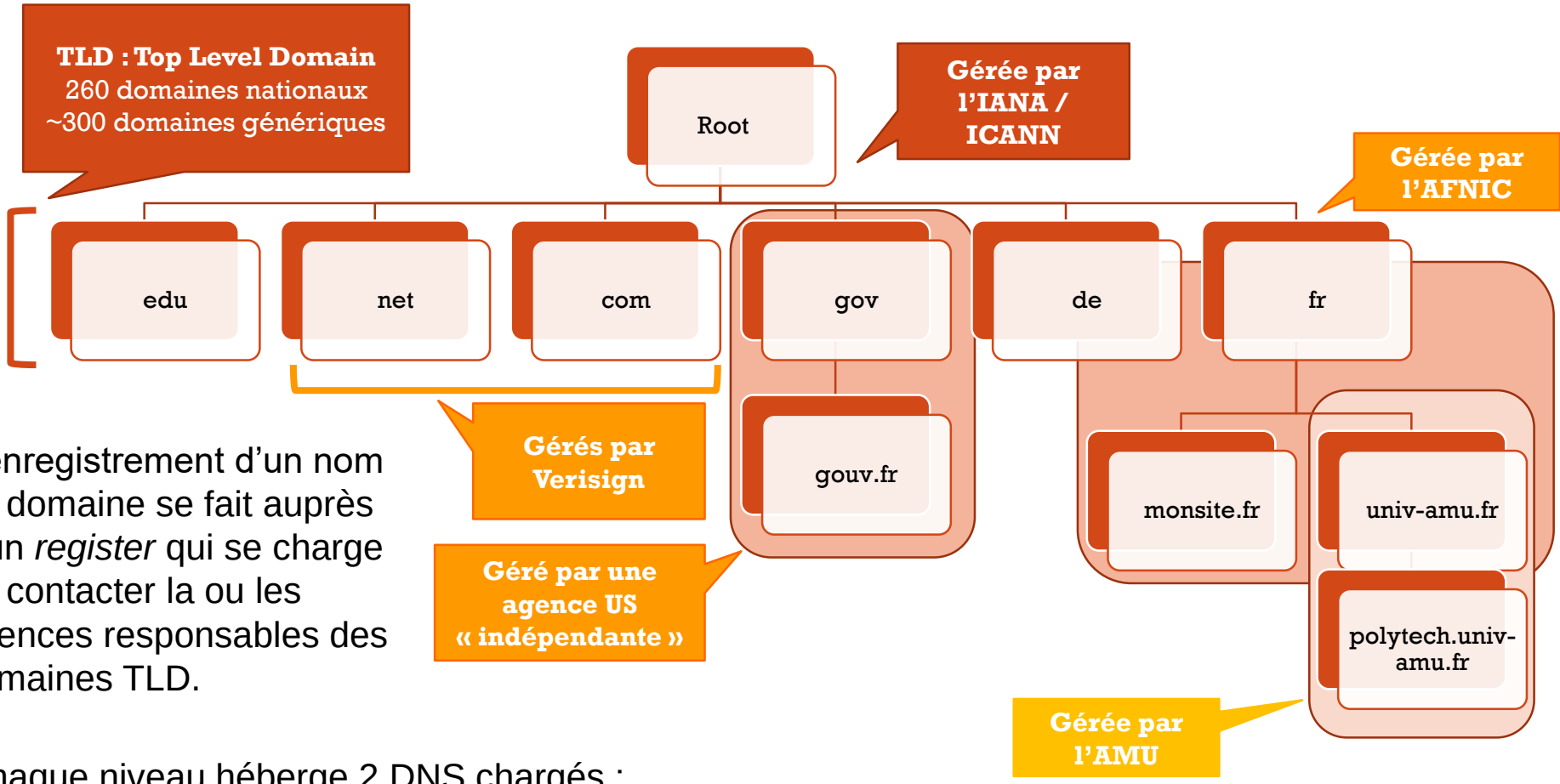
- Internet = *short for* réseau internet
  - => A.S.V.P.[a] = Tube / réseau logistique
- Web = World Wide Web
  - = contenu circulant dans le tube
  - = réseau/protocole IP + HTTP (protocole) + documents HTML (format de contenu)
  - Soit comme une navigateur sur une machine cliente va pouvoir récupérer la page « web » d'un serveur web (au USA par ex.).
- Autres contenu :
  - Mails (Protocoles SMTP & POP/IMAP)
  - Transferts fichiers (Protocole FTP)
  - Skype (protocole P2P propriétaire)
  - BitTorrent …

a) A.S. = Analogie Simpliste A Vocation Pédagogique, comparaison n'est pas raison.

PRINCIPE DU WEB…

- Norme
  - pour le transport (protocole IP)
  - pour l'échange de « messages » HTTP / HTTPS
  - et le format du contenu des « messages » HTML
- Indépendant des matériels connectés …
  - **Matériel** : Tablette/Ordinateur/Téléphone
  - **OS** (Système d'exploitation) : Mac/Linux/Windows
  - **Navigateur** (browser) : Firefox, IE Explorer, Chrome, Safari, Opera…
  - => le résultat doit être le même (obligation de respect des normes)
- … en théorie :
  - => http://browsershots.org/

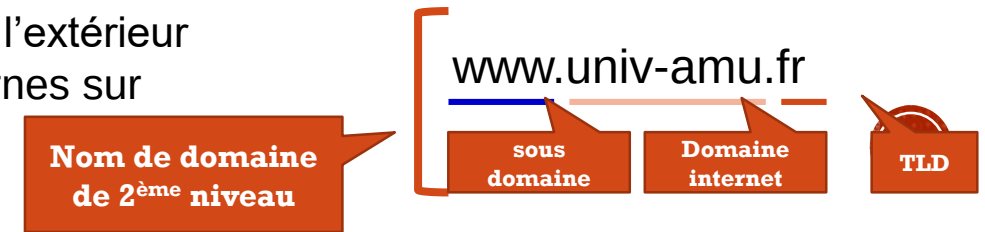
NOM DE DOMAINE INTERNET



L'enregistrement d'un nom de domaine se fait auprès d'un *register* qui se charge de contacter la ou les agences responsables des domaines TLD.

Chaque niveau héberge 2 DNS chargés :
- *En interne* : d'acheminer les requêtes DNS à l'extérieur
- *En externe* : de renseigner les requêtes externes sur l'adresse IP de la machine demandée

(ex. polytech.univ-amu.fr ou www.univ-amu.fr)



QUELQUES OUTILS DE BASE...

- Ex sous windows

Hote que l'on contacte

Machine (nom domaine & adresse IP) hébergeant l'hôte. Ici sous-domaine = www => serveur web

```
C:\Users>ping www.cnn.com

Envoi d'une requête 'ping' sur prod.turner.map.fastlylb.net
[151.101.60.73] avec
 32 octets de données :
Réponse de 151.101.60.73 : octets=32 temps=101 ms TTL=56
Réponse de 151.101.60.73 : octets=32 temps=99 ms TTL=56
Réponse de 151.101.60.73 : octets=32 temps=102 ms TTL=56
Réponse de 151.101.60.73 : octets=32 temps=99 ms TTL=56

Statistiques Ping pour 151.101.60.73:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 99ms, Maximum = 102ms, Moyenne = 100ms
```

Taille message envoyé

Temps de transport suivi de la valeur TTL restante

Moyenne calculée sur 4 envois

Trace pathping client

1. Calcul un chemin de la machine cliente vers l'hote (www.un.org)

```
C:\>pathping www.un.org

Détermination de l'itinéraire vers www.un.org [157.150.34.32]
avec un maximum de 30 sauts :
  0  LSIS-TRANVOUEZ.mshome.net [192.168.0.49]
  1  FREEBOX [192.168.0.254]
  2  far13-2-88-120-46-254.fbx.proxad.net [88.120.46.254]
  3  213.228.33.62
  4  marseille-6k-1-v806.intf.routers.proxad.net [212.27.51.117]
  5  p11-crs16-1-be1102.intf.routers.proxad.net [78.254.249.89]
  6  th2-9k-3-be1001.intf.routers.proxad.net [194.149.162.86]
  7  be4204.ccr21.par04.atlas.cogentco.com [149.11.115.13]
  8  be12308.ccr41.par01.atlas.cogentco.com [130.117.49.41]
  9  be2746.ccr41.jfk02.atlas.cogentco.com [154.54.29.117]
 10  be2056.ccr21.jfk10.atlas.cogentco.com [154.54.44.218]
 11  att.jfk10.atlas.cogentco.com [154.54.10.98]
 12  cr1.n54ny.ip.att.net [12.122.105.14]
 13  cr81.nw2nj.ip.att.net [12.122.105.30]
 14  ggr2.n54ny.ip.att.net [12.122.130.5]
 15  12.119.34.234
 16  157.150.192.240
 17     *        *        *
Traitement des statistiques pendant 400 secondes...
```

Hote que l'on contacte

IP Client (locale)

passerelle

Box Internet

Destination

TRACE PATHPING CLIENT

2. Ping chaque nœud du chemin afin d'identifier des problèmes réseaux

```
 Source vers ici   Ce noud/lien
Saut RTT    Perdu/Envoyé = % Perdu/Envoyé = % Adresse
 0                                            LSIS-TRANVOUEZ.mshome.net [192.168.0.49]
                                 0/ 100 =  0%   |
 1    5ms      0/ 100 =  0%      0/ 100 =  0%  FREEBOX [192.168.0.254]
                                 0/ 100 =  0%   |
 2   42ms      0/ 100 =  0%      0/ 100 =  0%  far13-2-88-120-46-254.fbx.proxad.net[88.120.46.254]
                                 0/ 100 =  0%   |
 3   59ms      0/ 100 =  0%      0/ 100 =  0%  213.228.33.62
                                 0/ 100 =  0%   |
 4   43ms      0/ 100 =  0%      0/ 100 =  0%  marseille-6k-1-v806.intf.routers.proxad.net[212.27.51.117]
                                 0/ 100 =  0%   |
 5   55ms      0/ 100 =  0%      0/ 100 =  0%  p11-crs16-1-be1102.intf.routers.proxad.net [78.254.249.89]
                                 0/ 100 =  0%   |
 6   55ms      0/ 100 =  0%      0/ 100 =  0%  th2-9k-3-be1001.intf.routers.proxad.net [194.149.162.86]
                                 0/ 100 =  0%   |
 7  103ms      0/ 100 =  0%      0/ 100 =  0%  be4204.ccr21.par04.atlas. cogentco.com [149.11.115.13]
                                 0/ 100 =  0%   |
 8  106ms      1/ 100 =  1%      1/ 100 =  1%  be12308.ccr41.par01.atlas.cogentco.com [130.117.49.41]
                                 0/ 100 =  0%   |
 9  174ms      0/ 100 =  0%      0/ 100 =  0%  be2746.ccr41.jfk02.atlas.cogentco.com [154.54.29.117]
                                 0/ 100 =  0%   |
    178ms      1/ 100 =  1%      1/ 100 =  1%  be2056.ccr21.jfk10.atlas.cogentco.com [154.54.44.218]
                                 0/ 100 =  0%   |
11   0ms      0/ 100 =  0%      0/ 100 =  0%  att.jfk10.atlas.cogentco.com [154.54.10.98]
                                 1/ 100 =  1%   |
12  ---       100/ 100 =100%    99/ 100 = 99%  cr1.n54ny.ip.att.net [12.122.105.14]
                                 0/ 100 =  0%   |
13  ---       100/ 100 =100%    99/ 100 = 99%  cr81.nw2nj.ip.att.net [12.122.105.30]
                                 0/ 100 =  0%   |
14  ---       100/ 100 =100%    99/ 100 = 99%  ggr2.n54ny.ip.att.net [12.122.130.5]
                                 0/ 100 =  0%   |
15  177ms      1/ 100 =  1%      0/ 100 =  0%  12.119.34.234
                                 1/ 100 =  1%   |
16  179ms      2/ 100 =  2%      0/ 100 =  0%  157.150.192.240

Itinéraire déterminé.
```

Pertes importantes de paquet

ARCHITECTURE CLIENT - SERVEUR

- Architecture basée sur une connexion réseau entre deux terminaux :
  - Le **client** : nécessitant un service / traitement / donnée => envoie une **requête** à un **serveur**.
  - Le **serveur** : responsable du traitement des requêtes ie prise en charge/exécution de la requête puis transmission du résultat.
- Le serveur :
  - écoute sur un port réseau dédié (ex. 80 pour HTTP) accessible par plusieurs clients.
  - Entame un dialogue avec le client selon le **protocole de communication** requis.

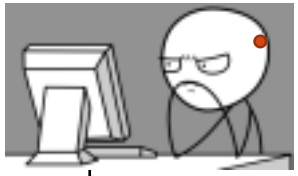
FONCTIONNEMENT D'UN SERVEUR WEB

- Serveur Web :
  - = application logiciel sur un serveur capable de comprendre des requêtes web (demande de pages html) et y répondre.
  - Utilise le protocole HTTP gérant la requête et l'envoi du résultat de la requête.
  - Le client demande une page web, le serveur lui renvoie la page (fichier) au format HTML (flux de caractères => format texte).
- Le client (un navigateur internet) fait des requêtes de fichiers HMTL à un serveur internet.

  ⇒Demander une information = demander 1 page dans laquelle l'information est déjà encodée en html

- HTML = langage de structuration de contenu
  - (on y reviendra)

FONCTIONNEMENT D'UN SERVEUR WEB (*OVER SIMPLIFIED*)



Aller sur
www.ici.net/index.html

Navigateur Web

Ordinateur Client

192.120.20.4

3 : protocole HTTP

4. Index.html

Serveur Web

Ordinateur Serveur

157.20.45.28

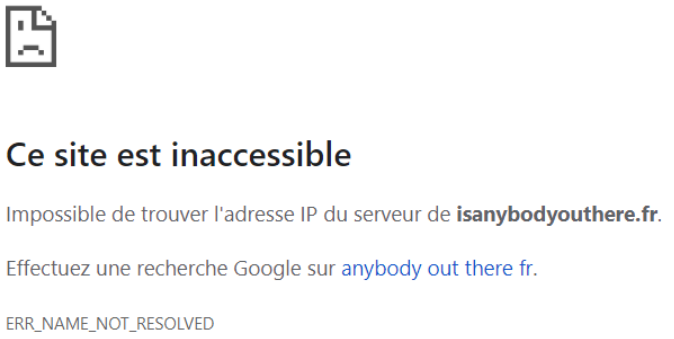
2. C'est l'IP 157.20.45.28

1. Adresse IP de www.ici.net

DNS

Domain Name Server : maintient une Table de correspondance : nom de Domaine / adresse IP

CENSURE PAR DNS



- Principe
  - Une autorité (judiciaire, politique) demande à ce qu'un site internet ne figure pas dans la base de donnée d'un ou plusieurs DNS (typiquement FAI)
  - Plus rigolo, indiquer une autre adresse IP que l'officielle : DNS Poisoning
- Exemples :
  - Blocage sites internets « pirates »
  - Blocage twitter en turquie (2015), youtube
- Solution ?
  - Utiliser un autre serveur DNS … qui peut aussi être ciblé (usurpation d'IP).

DNS de google… un ami ?

- Autres Techniques de censure :
  - Blocage IP (le routeur ne laisse pas passer les demandes vers une liste d'IP blacklistées)
  - Filtrage par paquet IP (on regarde le contenu du message et on laisse passer … ou non) : ex. filtrage contenus pear 2pear.

ILLUSTRATION PROTOCOLE HTTP

- Principe :
  1. Se connecter au serveur (port 80)
  2. Envoyer des commandes HTTP
- 2 commandes :
  - GET ressource : renvoi le contenu de la ressource ciblant un fichier contenant ou capable de produire du HTML
    - ex. GET / => envoi le fichier a la racine du site web cad en principe par défaut index.html

```
GET /page.html HTTP/1.0
Host: example.com
Referer: http://example.com/
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
```

  - POST : idem + envoi de données (formulaire) qui seront exploitées par le serveur

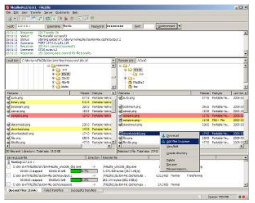
Illustration avec TELNET

EXEMPLE PAGE HTML

=> https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_document

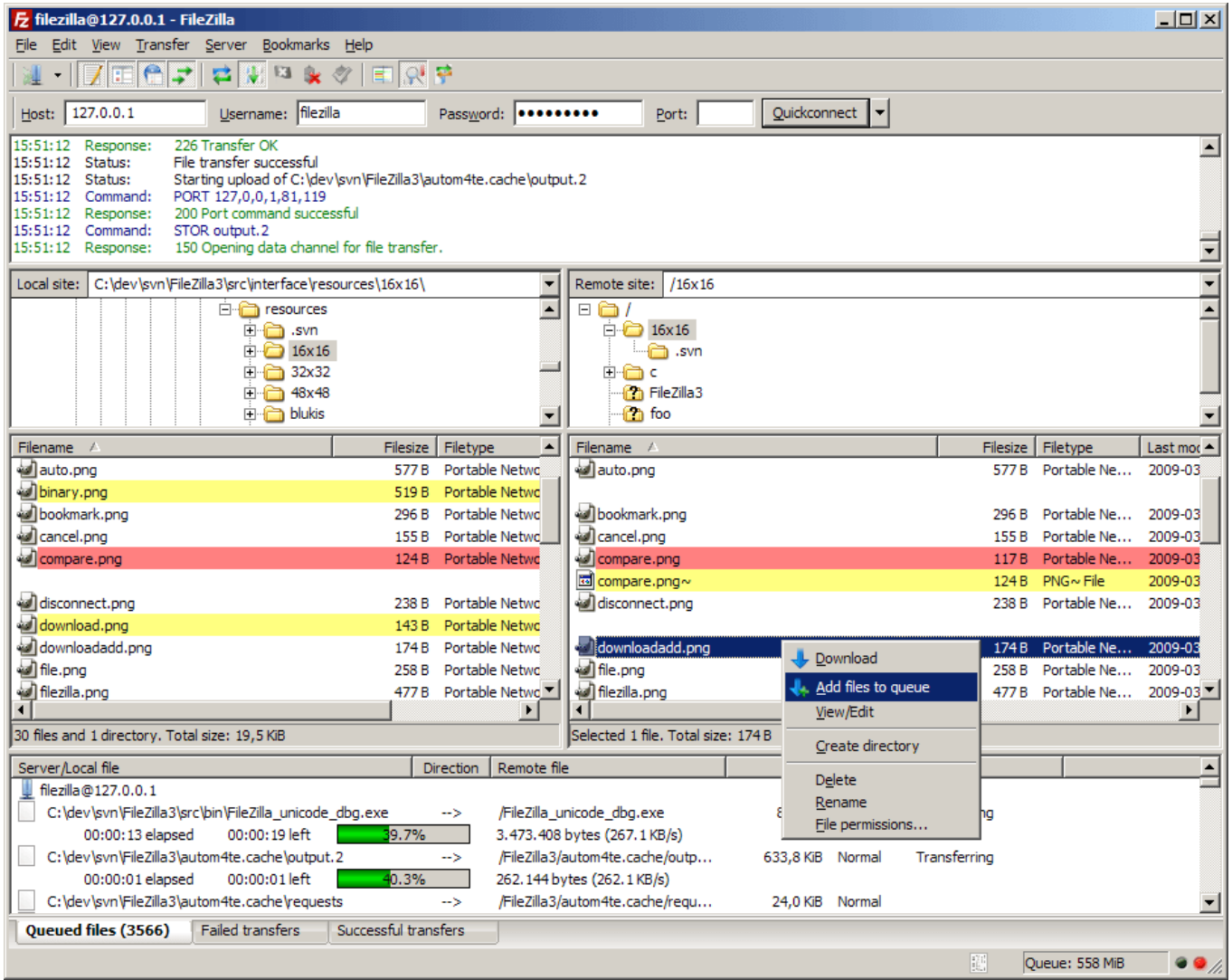
SITE WEB STATIQUE VS DYNAMIQUE

- Statique :
  - Les pages web ne s'adaptent pas au client… elles sont figées
  - Pour les mettre a jour il faut
    1. Editer le fichier en local
    2. uploader un nouveau fichier html (ftp par exemple)



- Dynamique :
  - La page web s'adapte au client selon les informations dont on dispose. Ex :
    - Site commerce en ligne : liste les commandes d'un client (après authentification). Le résultat (page web) sera différent selon les clients
    - Jeux en ligne MMO
  - 3 approches
    - Calculs fait depuis serveur (=> client ne voit que du html)
    - Calculs fait sur le client
    - Mixte (MMO)

EXEMPLE TRANSFERT FTP



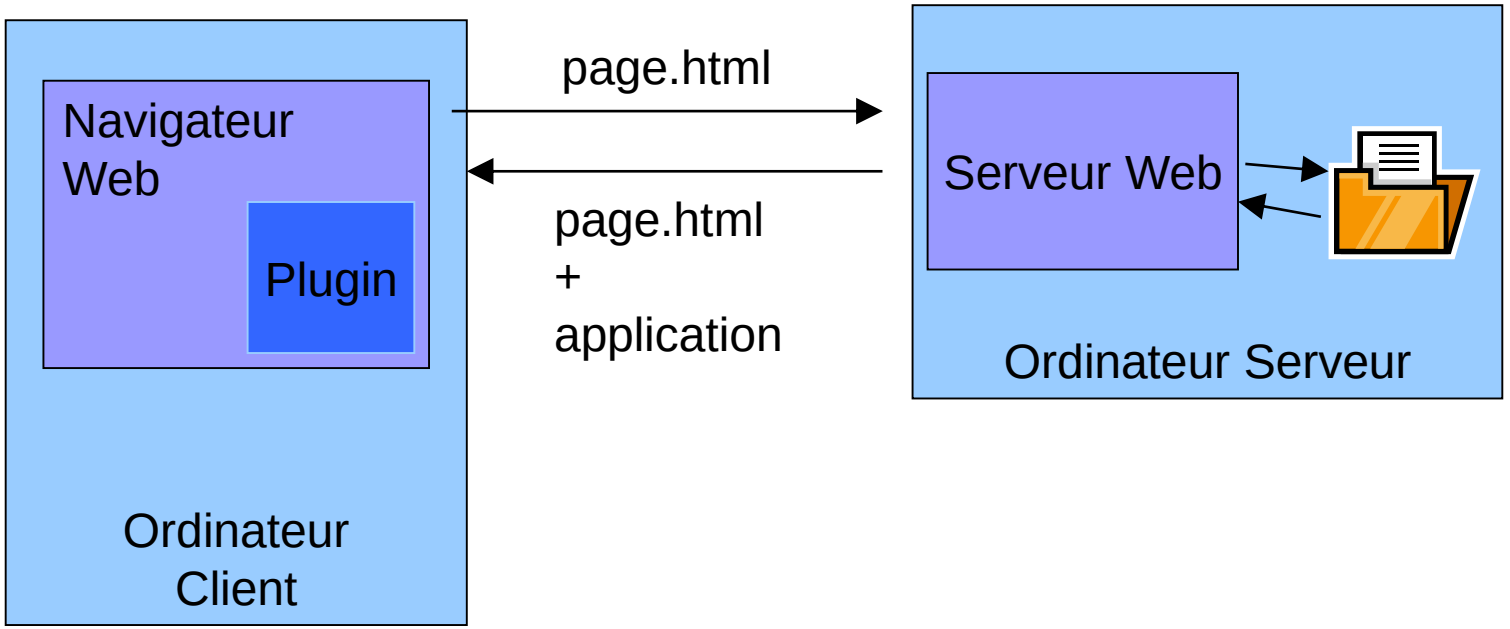
SOLUTION CÔTÉ CLIENT

- JavaScript :
  - Code téléchargé avec la page web et exécuté localement lors de la lecture de la page. L'information transformable doit donc déjà être disponible (ex. avec CSS modification de la visibilité d'un composant).
- Java :
  - Code intégré dans la page (aspect non HTML)
  - Possibilité d'exécuter des requêtes BD
  - Suppose l'existence d'une Machine Virtuelle Java (acquis)
  - Code compilé peut être lourd à télécharger
- Flash, etc…
  - Richesse de mise en forme (par rapport au débuts des applets)
  - Même inconvénient que Java
- HTML 5 !
  - Ajoute capacité de traitement (HTML 4 « passif »)
  - => exemple : lecteur video sans flash!

SOLUTIONS COTE CLIENT

⇒le navigateur web client dispose de toutes les informations pour répondre au besoin de l'utilisateur

⇒suppose le téléchargements d'informations non HTML

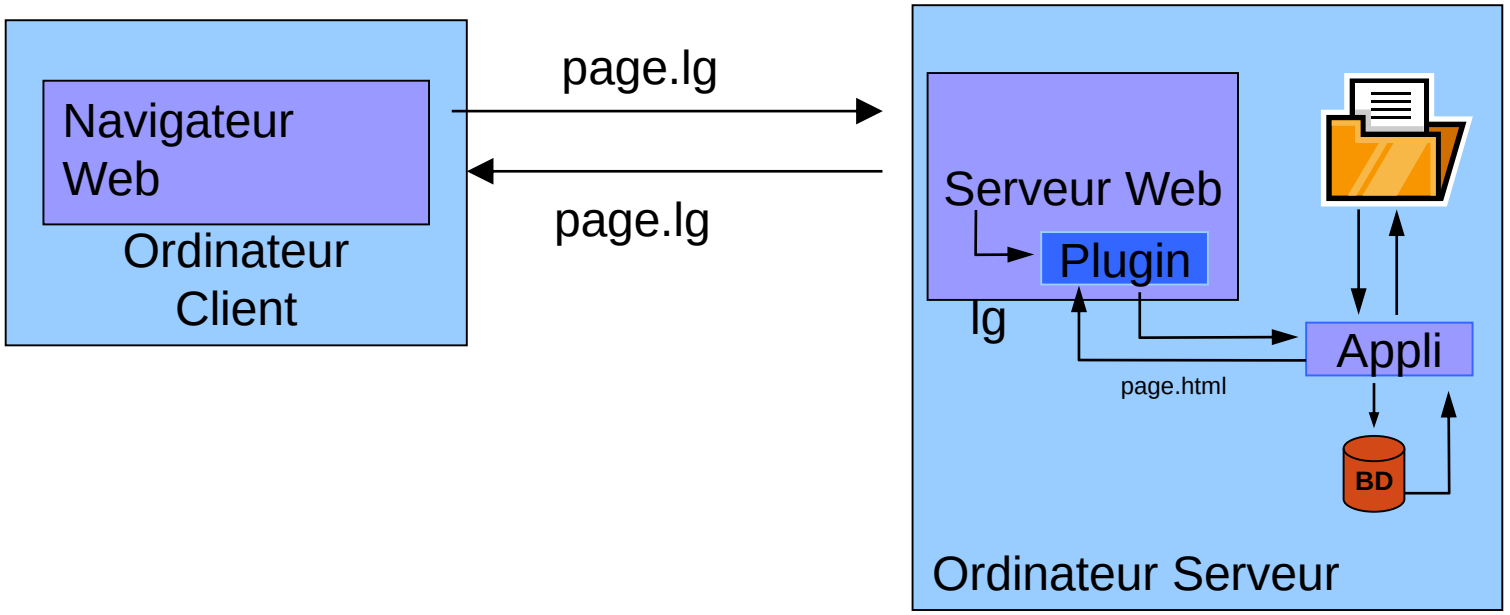


SOLUTION CÔTÉ SERVEUR

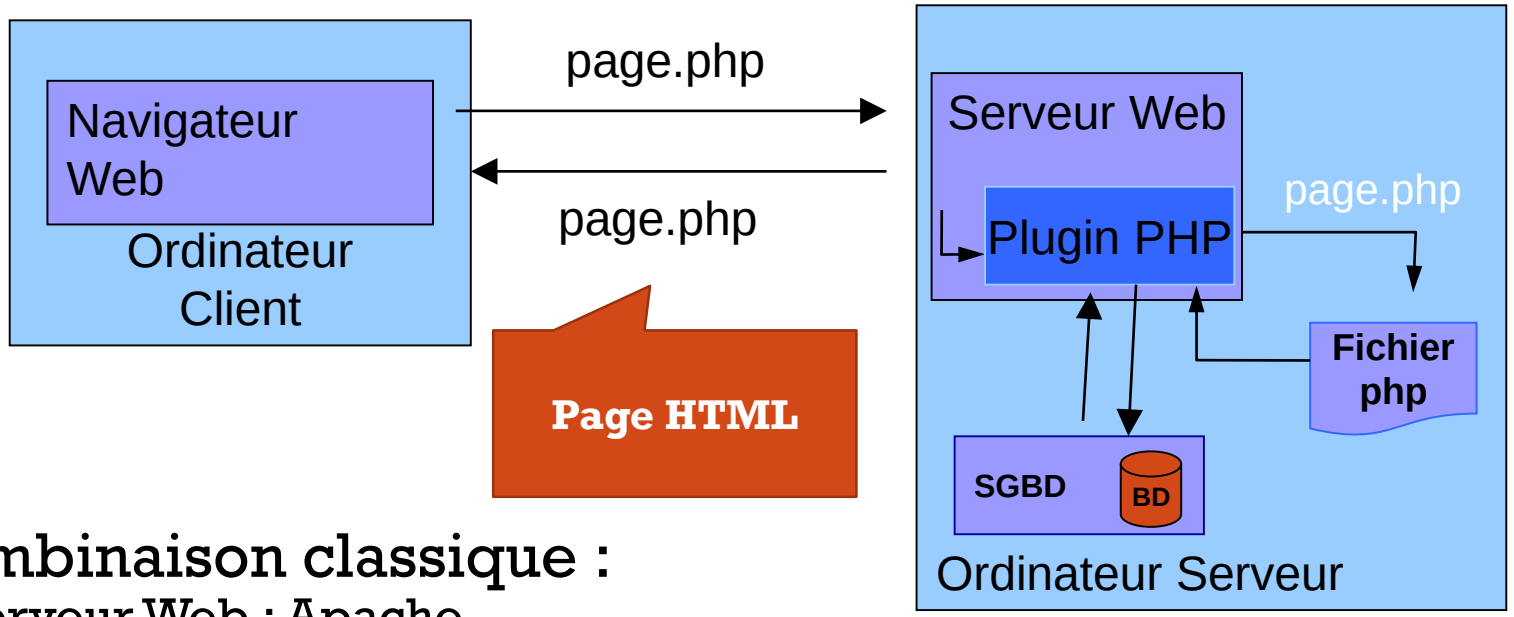
- C'est le serveur qui calcule la page HTML :
  - Le client ne reçoit qu'un fichier HTML =>pas de contraintes sur le client
- CGI (Common Gateway Interface) : le serveur transmet la requête HTML vers un programme dédié qui calcule et renvoie du code HTML au serveur
- Le programme peut être en
  - Code binaire (ie compilé) : ex. C, C++, Pascal
  - Langage de script : interprété par un programme dédié : ex. Bash, Perl, PHP, JSP, ASP…
- Suppose
  - que le serveur web est capable d'interagir avec ces programmes
  - une ressource avec capacité de calcul rapide (connexion avec plusieurs clients)
  - Donc application serveur optimisée

SOLUTIONS COTE SERVEUR

⇒le serveur calcule et renvoie l'information spécifique demandée par le client (sous forme HTML)



EXEMPLE DE COMBINAISON SERVEUR WEB & SGBD



- Combinaison classique :
  - Serveur Web : Apache
  - Module PHP dans Apache
  - SGBD : MySQL
- Exemple d'utilisation : banque => à vous !

SOLUTION HYBRIDE : MMO

- Coté client :
  - HTML + CSS + Javascript + PHP + Flash
- Côté Serveur :
  - Serveur dédié a la gestion des données utilisateurs/synchro entre joueurs
- Ex :

ARCHITECTURE HYBRIDE

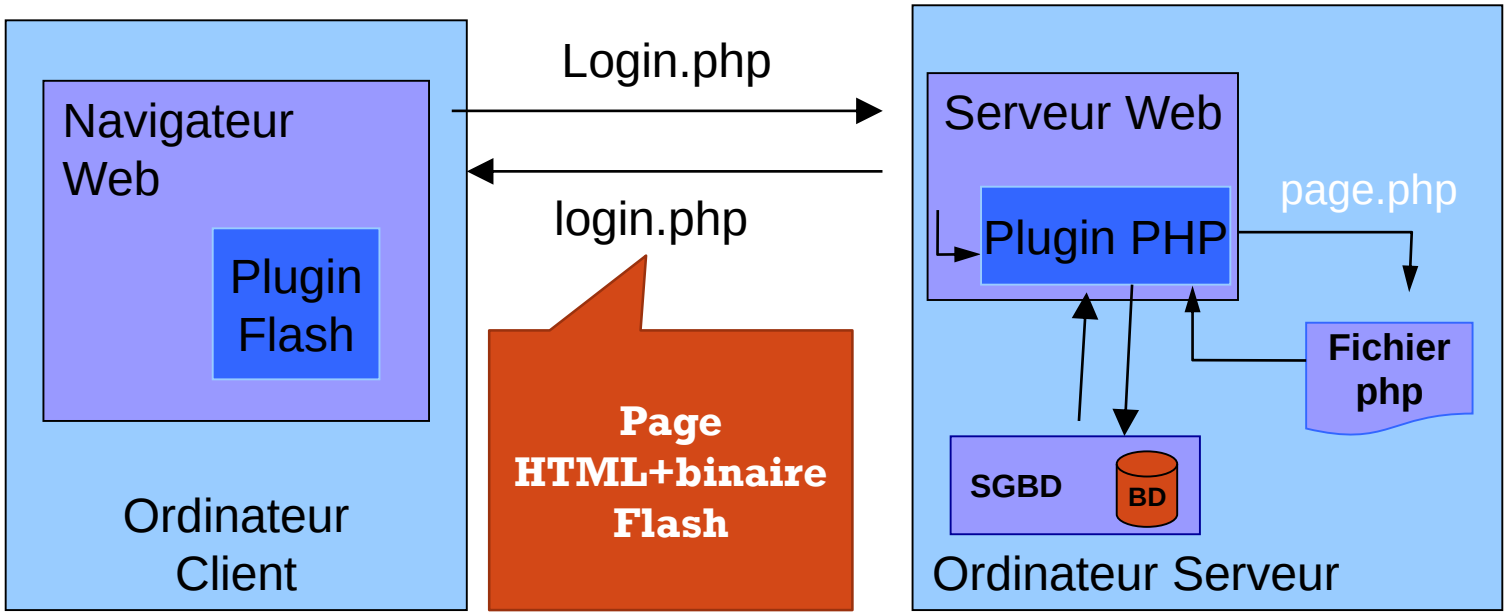
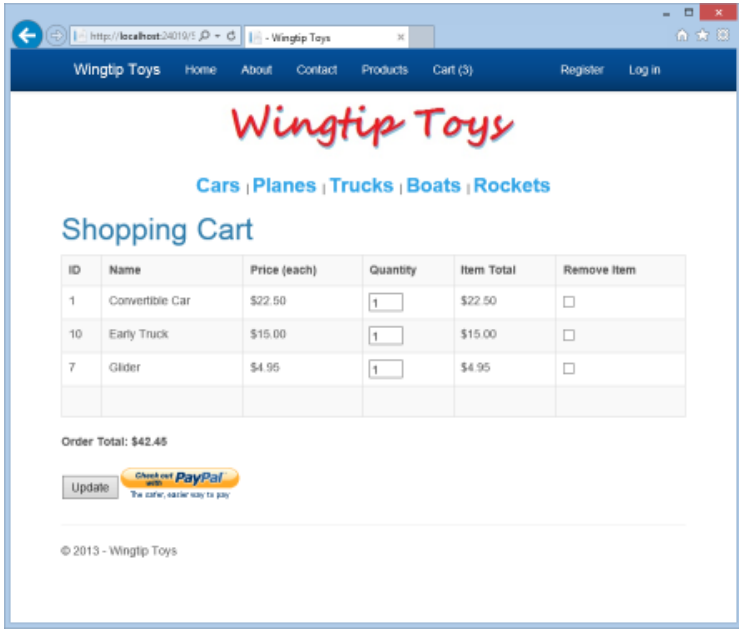


ILLUSTRATION : ÉVOLUTION COMMERCE EN LIGNE

- Capacité à voir un catalogue et à créer une commande
  - Solution HTML: page complète avec lien mail (au client d'indiquer ce qu'il veut)
  - Solution HTML+JavaScript : idem avec formulaire
  - Solution CGI :
    Programme C + HTML
    + JavaScript
  - Solution Scripts :
    Script (php) + HTML
    + JavaScript



RETOUR SUR HTML

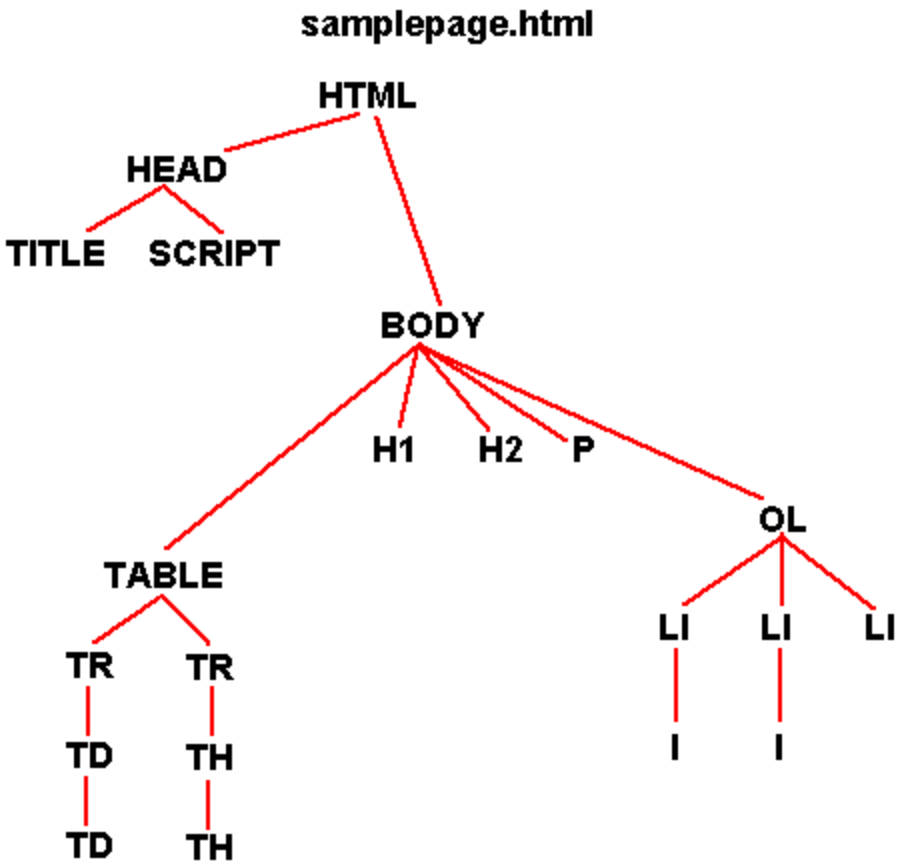
- HTML : Hypertext Markup Language
  - Créé par Tim Berners-Lee
  - Hypertext : capacité a créer des liens entre des documents
  - Markup = Balise
    - <TITLE> Ceci est un titre </TITLE>
  - Language : permet d'exprimer quelque chose. Implique
    - Un **vocabulaire** : mots « légitimes » => liste de balises définies, liste d'attributs, valeurs standards …
    - Une **grammaire** = syntaxe : agencement légitime de mots (ouverture balise, fermeture balise, non recouvrement de balises …)

EXEMPLE DE BALISES…

- La catégorie du texte est définie par les 2 balises qui l'encadrent :
  - <Balise>Texte contenu balise</Balise>
- Balise générales
  - Titre : TITLE
  - Corps de la page : BODY
- Sous catégorie
  - Niveau de Titres : H1, H2,… Hn
  - Paragraphe : P
  - Citation : QUOTE
  - Liste avec puce : UL (unordered list)
  - Liste numérotée : OL (ordered list)
  - Item de liste : IL
  - …

RAPPEL DES CATÉGORIES DE TEXTE HTML

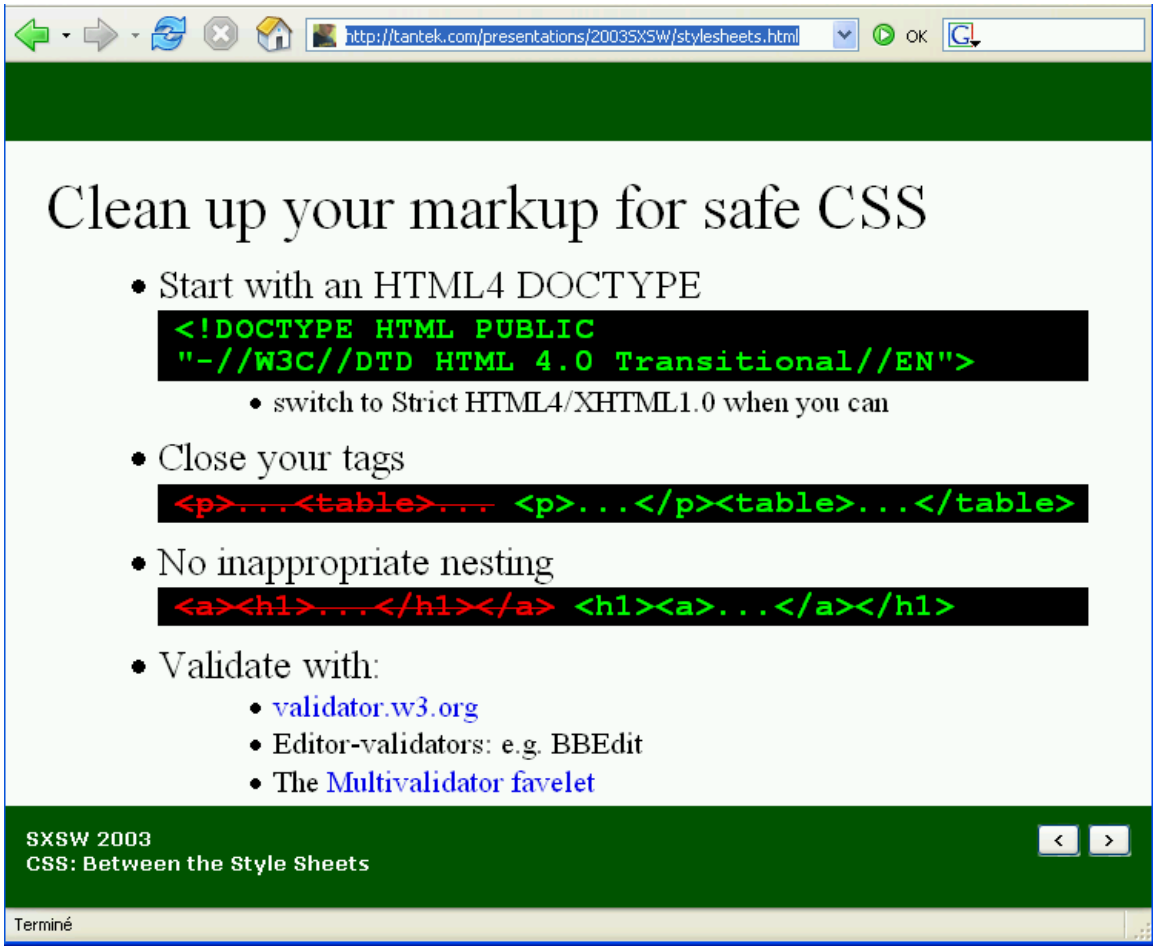
- Il y a une hiérarchie des balises …
- Qui donne un sens au contenu



QUE VA-T-ON VOIR

- HTML
  - Utilisation des différentes balises
  - Relativement simple
- CSS
  - Mise en forme des pages web indépendamment du contenu (HTML)
  - Plus complexe (notamment héritage d'attributs)
- Javascript
  - Ajout de capacité de traitement sur une page web
  - Programmation !

EXEMPLE CSS – PRÉSENTATION POWERPOINT ?



http://tantek.com/presentations/2003SXSW/stylesheets.html

NOPE : HTML+CSS+JAVASCRIPT !

EXEMPLE – LE MÊME SANS LES STYLES !

+ un peu de javascript :)